Effective Authorship Attribution in Large Document Collections

A thesis submitted for the degree of
Doctor of Philosophy

Ying Zhao, B.CompSci
School of Computer Science and Information Technology,
Science, Engineering, and Technology Portfolio,
RMIT University,
Melbourne, Victoria, Australia.

December 20, 2007

Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged.

Ying Zhao
School of Computer Science and Information Technology
RMIT University
December 20, 2007

Acknowledgments

I would like to thank many people who gave me support to complete this thesis. Without their support this thesis would not appear in its present form.

Any words would be too plain to express my deep and sincere gratitude to my amazing primary supervisor, Prof. Justin Zobel. He taught me how to think critically, how to enjoy research, and how to be a good researcher. Without his enthusiasm, his inspiration, his encouragement, and his great efforts to guide me throughout my study, I would never have finished. He taught me to be positive and brave in a hard life; without his support and encouragement, I would have quit my studies when my family crisis happened. He is a great supervisor.

Besides, I would like to thank my second supervisor, Dr. Phil Vines, for his guidance and assistance. My heartfelt thanks are also made to Dr. Falk Scholer and Steven Garcia for proof reading this thesis and valuable feedbacks.

Also, I thank all the students in the Search Engine Group at RMIT University, who filled my PhD life with laughter and happiness.

Finally, I would like to express special thanks to my parents for their love, understanding, and encouragement in every possible way. To me, my Mum and Dad are the greatest in the world.

Preface

Portions of the material included in this thesis have previously appeared in the following publications:

- "Entropy based Authorship Search in Large Document Collections", in *Proceedings of* **29th ECIR** *Twenty-ninth European Conference on Information Retrieval* [Zhao and Zobel, 2007b].
- "Authorship Attribution via Combination of Evidence", in *Proceedings of* **29th ECIR** *Twenty-ninth European Conference on Information Retrieval* [Zhao and Vines, 2007].
- "Search with Style: Authorship Attribution in Classic Literature", Winner, the **best student paper**, in *Proceedings of* **30th ACSC** *Thirtieth Australia Computer Science Conference* [Zhao and Zobel, 2007a].
- "Using Relative Entropy for Authorship Attribution", in *Proceedings of* **3rd AIRS** *Asia Information Retrieval Symposium* [Zhao et al., 2006].
- "Effective and Scalable Authorship Attribution Using Function Words", in *Proceedings of* **2rd AIRS** *Asia Information Retrieval Symposium* [Zhao and Zobel, 2005].

Contents

Abstract **1**

1 Introduction **3**

1.1 Authorship Attribution 4

1.1.1 Issues Remaining in Authorship Attribution 10

1.2 Research Contributions 11

1.3 Thesis Structure 13

2 Background **15**

2.1 Text Categorization 16

2.1.1 Benchmarks in Text Categorization 18

2.1.2 Document Representation 20

2.1.3 Feature Weighting in Text Categorization 21

2.2 Machine Learning for Text Categorization 22

2.2.1 Data Overfitting 23

2.2.2 Avoiding Overfitting 24

Train-and-Test 24

Cross validation 25

2.2.3 Evaluations in Text Categorization 25

2.2.4 Text Classifiers 28

Naïve Bayesian Classifiers 29

Bayesian Network Classifiers 30

Nearest Neighbour & K-Nearest Neighbour Classifiers 35

Decision Tree Classifiers . . . 37
Support Vector Machines . . . 40
2.3 Authorship Attribution . . . 46
2.3.1 Stylometry . . . 47
Lexical Features . . . 49
Linguistic Features . . . 50
2.3.2 Collections in Authorship Attribution . . . 53
2.4 Existing Approaches for Authorship Attribution . . . 56
2.4.1 Simple Statistics Measures . . . 56
2.4.2 Principal Component Analysis . . . 58
2.4.3 N-grams and Markov Chains . . . 61
2.4.4 Compression Techniques . . . 65
2.4.5 Machine Learning Approaches . . . 67
2.5 Chapter Summary . . . 69

3 Collections and Preliminary Investigation 71
3.1 Developing Good Test Collections . . . 72
3.1.1 The Associated Press . . . 73
Eliminating Near-duplicate Documents . . . 74
Standardising Inconsistencies in Authorship . . . 74
3.1.2 Project Gutenberg . . . 79
3.2 Preliminary Investigation . . . 81
3.2.1 Stylometric Features . . . 81
3.2.2 First Try: Principal Component Analysis . . . 82
3.2.3 Baseline Experiments . . . 86
Binary Authorship Attribution: With Weka . . . 88
Binary Authorship Attribution: With SVMs . . . 92
Multi-class Authorship Attribution . . . 93
Side Experiments: The Federalist Papers . . . 95
One-class Authorship Attribution . . . 96
Computational Cost . . . 99

Refinement of one-class Experiments 101
3.3 Chapter Summary 105

4 Relative Entropy for Authorship Attribution 107
4.1 Background 108
4.2 Information Theory and Entropy 110
4.3 Relative Entropy: Kullback-Leibler Divergence 111
4.4 Language Models and Smoothing 113
4.4.1 Absolute Discounting 116
4.4.2 Jelinek-Mercer Smoothing 116
4.4.3 Dirichlet Prior Smoothing 116
4.4.4 Two-Stage Smoothing 117
4.5 KLD as a Classifier for Authorship Attribution 117
4.6 Experiments 119
4.6.1 Binary Authorship Attribution 120
Smoothing Effectiveness 121
KLD versus Other Methods 125
4.6.2 Multi-class Authorship Attribution 127
4.6.3 KLD as a Classifier for Text Categorization 128
4.7 Chapter Summary 129

5 Style Markers in Authorship Attribution 132
5.1 Style Markers 133
5.2 First Results: Individual Marker Types 138
5.3 Authorship Attribution via Combination of Evidence 141
5.3.1 Model Voting System 143
5.3.2 Two-Stage Model Prediction System 145
5.3.3 Additive Modelling System 146
5.4 Experiments and Results 147
5.5 Chapter Summary 154

6 Authorship Search **156**

6.1 Motivation . . . 157

6.2 Methods of Authorship Attribution . . . 158

6.3 Document Search . . . 158

6.3.1 Vector Space Model . . . 159

6.3.2 Probabilistic Models . . . 162

6.3.3 Language Models . . . 163

6.4 Style-based Authorship Search . . . 164

6.4.1 Indexing Strategy . . . 164

6.4.2 Entropy-based Similarity Measure for Authorship Search . . . 166

6.5 Experiments: Authorship Search . . . 169

6.5.1 Feasibility and Scale in Size . . . 170

6.5.2 KLD Ranking versus Other Measures . . . 174

6.5.3 Index with Different Style Markers . . . 175

6.5.4 Applicability to Authorship Attribution . . . 179

6.6 Chapter Summary . . . 184

7 Authorship Attribution in Classic Literature **186**

7.1 Background . . . 186

7.2 Experiments and Results . . . 188

7.2.1 Testbed: Gutenberg634 . . . 188

7.2.2 Indexing Mechanism . . . 190

7.2.3 Classification-based Authorship Attribution . . . 191

7.2.4 Authorship Search . . . 195

7.2.5 Shakespeare and His Contemporaries . . . 199

7.2.6 Beyond Precision: Authorship Search for Authorship Attribution . . . 203

7.3 Summary . . . 204

8 Conclusions and Future Work **206**

8.1 Research Contributions . . . 207

8.2 Future Work . . . 215

A The List of Selected Function Words **218**

B The List of Selected Brown Tags **221**

Bibliography **225**

List of Figures

2.1 Relationship between precision and recall 27
2.2 Network structure of a naïve Bayesian classifier 32
2.3 An example of a Bayesian network 33
2.4 An example of a k-NN classifier 36
2.5 An example of a decision tree 39
2.6 An example of a SVM for classification 40
2.7 An example of linear separating hyperplane in SVMs 42
2.8 An example of non-linear separable data in SVMs 45
2.9 An example of grammatical structure of a sentence 51

3.1 The distribution of document length in words for different authors 78
3.2 Examples of applying PCA to binary AA 84
3.3 Examples of applying PCA to 3-class AA 85
3.4 Scalability of N-class AA in the number of authors 94
3.5 Scalability of one-class AA in the number of negative samples 98
3.6 Refinement of one-class AA with 25 positive samples 103
3.7 Refinement of one-class AA with 300 positive samples 104

4.1 Illustration of smoothing applied in IR 118
4.2 Illustration of smoothing applied in AA 119
4.3 Effectiveness of smoothing all pre-defined features for AA 122
4.4 Effectiveness of query-centric smoothing for AA 122

5.1 An example of a syntax tree 136
5.2 An example of 3-class AA using the KLD attribution model 142
5.3 An example of 4-class AA using the KLD attribution model 142
5.4 Framework for the model voting system 144
5.5 Framework of the two-stage model prediction system 145
5.6 Framework of the additive modelling system 146
5.7 An example of predicting the best model for 3-class AA 150

6.1 An example of index terms used for AS 165
6.2 Scalability of AS in the size of the collections 173
6.3 Scalability of AS in the volume of the queries 173
6.4 Comparison of different similarity measures for AS 174
6.5 Comparison of indexing topic-free terms and top-bearing terms for AS 175
6.6 Evaluation of using different marker types for AS on AP10k 176
6.7 Evaluation of using different marker types for AS on AP100k 176
6.8 Evaluation of using different marker types for AS on AP500k 176

List of Tables

1.1 Comparison between traditional AA and non-traditional AA 9

2.1 The number of documents in the top 8 categories in Reuters-21578 19
2.2 Components used for evaluation in text categorization 26
2.3 An example of rewrite rules for AA 52
2.4 Comparison of compression programs for AA 66

3.1 Examples of multiple name representations for an author 75
3.2 Examples of possible typographic errors in the author names 75
3.3 Statistics of the AP7 collection 76
3.4 An example of usage statistics for common function words 82
3.5 Effectiveness of binary AA using 5 machine learning methods and 10-fold cross validation 88
3.6 Effectiveness of binary AA using 5 machine learning methods and train-test evaluation 89
3.7 Effectiveness of binary AA on an author-by-author basis 90
3.8 Paired t-test between different methods with binary AA 91
3.9 Effectiveness of SVMs for binary AA 92
3.10 Effectiveness of multi-class AA using 5 machine learning methods 93
3.11 The investigation with the Federalist Papers 96
3.12 Effectiveness of one-class AA using 5 machine learning methods 97
3.13 Paired t-test between different methods with one-class AA 100
3.14 Comparison in efficiency between methods 101

3.15 An example of statistics of frequently used function words 102

4.1 Effectiveness of smoothing methods for binary AA on AP7 123
4.2 Effectiveness of smoothing methods for binary AA on GutenbergSmall 123
4.3 Paired t-test on AP7 between different smoothing methods 124
4.4 Comparison between Bayesian networks, SVMs, and KLD attribution model for binary AA on AP7 125
4.5 Paired t-test between KLD and SVMs on AP7 126
4.6 Comparison between Bayesian networks, SVMs, and KLD attribution model for binary AA on GutenbergSmall 127
4.7 Effectiveness of Bayesian networks and KLD attribution model for multi-class AA 128
4.8 KLD attribution model for general text categorization 130
4.9 A comparison between the KLD attribution model and SVMs for general text categorization 130

5.1 Effectiveness of single type of style marker for AA 140
5.2 Effectiveness of the model voting system for binary AA 148
5.3 Effectiveness of the two-stage model prediction system for AA 151
5.4 Effectiveness of additive modelling system using two feature models 152
5.5 Effectiveness of the additive modelling system using more than two feature models 153
5.6 Paired t-test for the additive modelling system 154

6.1 The atomic components used in similarity measure in IR 160
6.2 Summary of notations in KLD framework for AS 167
6.3 The number of correct matches in the top 100 ranked documents in AS . . . 171
6.4 The number of correct matches in the top 100 ranked documents in AS, using 100-included document queries 171
6.5 Comparison between the authors: Dishneau and Beamish 178
6.6 Effectiveness of search-based AA on APvote10k 181
6.7 Effectiveness (author specific) of search-based AA on APvote10k 182

6.8 Effectiveness (author specific) of search-based AA on APvote100k 183

7.1 Strong inconsistency of documents from Project Gutenberg 190
7.2 Usage statistics for common function words for Shakespeare and Marlowe . . 191
7.3 Results (better than 90% on function words) of one-class AA 193
7.4 Results (less than 90% on function words) of one-class AA 194
7.5 Results of p@5 (greater than 80% on function words) of search-based AA . . 197
7.6 Results of p@5 (less than 80% on function words) of search-based AA 198
7.7 Example ranked lists for works of Shakespeare 200
7.8 Example ranked lists for works of Beaumont & Fletcher 201
7.9 Example ranked lists for works of Marlowe 201
7.10 Example ranked lists for works of Jonson 201

List of Algorithms

1 KLD exhaustive ranking algorithm for AS 169
2 KLD search-based algorithm for AA 179

Abstract

Techniques that can effectively identify authors of texts are of great importance in scenarios such as detecting plagiarism, and identifying a source of information. A range of attribution approaches has been proposed in recent years, but none of these are particularly satisfactory; some of them are ad hoc and most have defects in terms of scalability, effectiveness, and computational cost.

Good test collections are critical for evaluation of authorship attribution (AA) techniques. However, there are no standard benchmarks available in this area; it is almost always the case that researchers have their own test collections. Furthermore, collections that have been explored in AA are usually small, and thus whether the existing approaches are reliable or scalable is unclear. We develop several AA collections that are substantially larger than those in literature; machine learning methods are used to establish the value of using such corpora in AA. The results, also used as baseline results in this thesis, show that the developed text collections can be used as standard benchmarks, and are able to clearly distinguish between different approaches.

One of the major contribution is that we propose use of the Kullback-Leibler divergence, a measure of how different two distributions are, to identify authors based on elements of writing style. The results show that our approach is at least as effective as, if not always better than, the best existing attribution methods—that is, support vector machines—for two-class AA, and is superior for multi-class AA. Moreover our proposed method has much lower computational cost and is cheaper to train.

Style markers are the key elements of style analysis. We explore several approaches to tokenising documents to extract style markers, examining which marker type works the best.

We also propose three systems that boost the AA performance by combining evidence from various marker types, motivated from the observation that there is no one type of marker that can satisfy all AA scenarios.

To address the scalability of AA, we propose the novel task of authorship search (AS), inspired by document search and intended for large document collections. Our results show that AS is reasonably effective to find documents by a particular author, even within a collection consisting of half a million documents. Beyond search, we also propose the AS-based method to identify authorship. Our method is substantially more scalable than any method published in prior AA research, in terms of the collection size and the number of candidate authors; the discrimination is scaled up to several hundred authors.

Chapter 1

Introduction

Writing style is an approach to the construction of sentences. Writing can describe events in many ways, as prose or verse; expressions can be organized in either passive voice or active voice; descriptions can be tedious, elaborate, or concise but precise; content can be easy to follow or difficult. Choices of these sentence constitutions are diverse from author to author, and reflect different styles of writing.

Some writers prefer using short sentences to convey straightforward meaning to readers, while others may often choose complicated grammar to constitute extremely long sentences with complex structures by; for instance, adding semi-colons and clauses. On the other hand, even with the same author, the writing style—which may reveal the personalities, thoughts, and voices in his or her productions—may be influenced or changed by educational background and life experience.

The notion of style is central to literature. The best-known authors of classic English novels and plays are renowned for having distinctive styles that make their works immediately recognizable. For example, William Shakespeare (1564–1616) has gained worldwide popularity as the greatest writer of the English language and the world's preeminent dramatist, having written approximately 37 plays and 154 sonnets, as well as a variety of other poems. Shakespeare's plays were written in verse, not prose. Neologism is another distinct characteristic of his style. Neologism refers to a made-up word that is not a part of normal or everyday vocabulary. Some scholars have suggested that Shakespeare has used around 17,677 distinct words in his works, with approximately 10% neologisms. These new items were sometimes

borrowed from classical literature or foreign languages; as an example, he invented the word "climature" as a mix between "climate" and "temperature". Some researchers also suggested that Shakespeare followed certain rules in selecting numbers of syllables that should be included in each sentence. Henry James (1843–1916) was a prolific writer who authored many novels, short stories, and essays on a variety of topics. James's style is regarded as difficult, elaborate, and obscure, relying heavily on extremely long sentences, by deferring the verbs, including many prepositional phrases and subordinate clauses. Charles Dickens (1812–1870) is widely considered as one of the greatest novelists in the English language, and is famous for the humour in his works. Mark Twain (1835–1910), another renowned writer, had a distinctive writing style that is a mixture of humour, irony, and usage of American idioms.

A reader who is familiar with particular novelists can easily recognize their writing. This suggests that, to a certain extent, the writing style can be differentiated between authors, and thus can be an indication of authorship. Stylometry is concerned with the study of linguistic styles, usually in written language. It is often used to attribute authorship to anonymous or disputed documents, when the author information is missing, doubtful, or controversial. The study of stylometry shows that it is feasible to undertake authorship attribution—which has legal, academic, and literary applications—ranging from the question of the authorship of Shakespeare's works to forensic linguistics.

1.1 Authorship Attribution

Authorship attribution (AA), as the name implies, is the task of identifying who wrote a particular document, by analysing the writing style of that document. It has a wide range of applications. Academics use AA to analyse differences in writing style of the anonymous or disputed documents in literature—such as the plays of Shakespeare, and the twelve disputed Federalist Papers [Fung, 2003; Juola, 1997; Khmelev and Tweedie, 2002]—to attempt to pick out the actual authors of these texts.

Plagiarism detection is a potential application. AA can be used to establish whether claimed authorship is valid, and may be able to determine the origin of a piece of text when there are several authors claiming authorship. Managing plagiarism has been a challenge in academic departments. In a university environment, some students plagiarize essays or

assignment solutions from each other. Also, when there is little change in an assignment specification over the time, it is not hard for current students to get solutions from former students. In addition, it is easy to copy and paste materials from the web. In these situations, AA has obvious value.

In forensic investigations, AA can be applied to verify origin of e-mails and posts in newsgroups [Koppel and Schler, 2003]. It is particularly helpful for verifying identities of suspicious activities in use of computers, such as illegitimate email usage. Also, AA can potentially be used to identify the source of a piece of intelligence[Carole, 2005]. Moreover, AA may contribute to the investigation of crime.

AA studies are diverse; the categorization of AA tasks can be made in different dimensions. Based on the number of candidate authors involved, AA investigations can be divided broadly into five categories as described below.

Binary authorship attribution.

Binary AA is simplest type of AA, in which only two author candidates are considered. The underlying assumption in this kind of AA is that, the anonymous documents to be identified are written by one of the two candidates. For example, a study investigating who wrote the doubtful drama *Macbeth*—Shakespeare or Bacon—is a typical binary AA problem. Many early AA investigations are case studies, focusing on certain authors; most of these studies are binary AA.

Multi-class authorship attribution.

Multi-class AA is also referred as n-class AA, where $n > 2$. For example, identifying who wrote *Macbeth*—Shakespeare, Marlowe or Bacon—is a 3-class AA problem. In binary AA the effectiveness of attribution is 50% at random, however in n-class AA, it is $1/n$. Clearly, the more potential authors provided, the harder the task is.

Authorship verification.

In authorship verification, there is only one potential author provided; therefore, it is sometimes known as one-class authorship attribution. The purpose is to verify the validity of this

potential author, for instance Shakespeare, for an disputed work, such as *Macbeth* (according to some). Authorship verification can be used to examine whether *Macbeth* is written by Shakespeare. Effective authorship verification relies on the prior knowledge of a potential author, Shakespeare, and the discrimination is made between Shakespeare and any other (non-Shakespeare) authors.

Authorship identification.

Authorship identification determines the author of an anonymous text when there is no potential author specified. In this situation, the identification mainly relies on domain experts and linguistic experts. However the results are subject to prior knowledge; different experts may have different theories to support their findings. It is not feasible to undertake this kind of identification by an automated system, and therefore it is not investigated in this thesis.

Authorship collaboration.

Authorship collaboration is concerned with whether a document is solely written by a certain author, or collaboratively composed by a few authors. Authorship collaboration is not widely explored due to the limitation of data resources, and is not covered in this dissertation.

Apart from the categorization mentioned above, AA studies can be grouped in another dimension—that is, traditional AA and non-traditional AA.

Traditional Authorship Attribution

Traditional AA employs both internal and external evidence to determine the author of a given text. It often deals with disputed texts in literature. Internal evidence refers to information that can be collected from the text itself. Typical examples are: words and part-of-speech tags that are frequently used; length of words, sentences, or documents; and structures of sentences. External evidence relies on domain experts who are expert in the field of the work being analysed, such as biographers of the disputed authors.

In general, traditional AA requires substantial human effort. Exhaustive biographical investigations of potential authors are of great importance; these investigations are from many viewpoints, including authors' educational level, life experience, life attitude, and personality,

as well as family background. Some authors may change styles of writing between time periods in their lives. Besides, style is subjective; different readers have different interpretations of what kinds of writing style a document has. All of the five attribution tasks listed above can be undertaken in the traditional way.

The Shakespeare authorship problem[1] is a typical example that illustrates traditional AA. Around 150 years after Shakespeare's death in 1616, doubts began to be expressed by some researchers regarding the authorship of the plays and poetry that have been attributed to him. Many scholars argue that William Shakespeare was the actual name of the author for all the works. However on the other hand, many researchers believe the works to have been written by another playwright. The disagreement arose due to the lack of biographical evidence that could support the authorship of Shakespeare's works. There are 33 commonly noted arguments[2] against the attribution of authorship to Shakespeare, from various perspectives that are mainly related to external evidence. Several candidates were suggested as the potential true authors who should be attributed to Shakespeare works.

Edward de Vere, the 17th Earl of Oxford, remains the most prominent alternative candidate for authorship of the Shakespeare canon. It was based on his literary reputation, educational level, as well as similarities between the Earl's life and events depicted in the plays and sonnets. Christopher Marlowe and Francis Bacon are two further alternatives who have been brought into the argument. Marlowe was regarded as the foremost Elizabethan tragedian before Shakespeare. It has been speculated that Marlowe's recorded death in 1593 was faked, and that he subsequently wrote under the name of William Shakespeare. Francis Bacon is selected as his travel experience is similar to that described in Shakespeare's works. Academics continue to attempt to attribute the authorship of plays and poems by traditional means, for both those attributed to Shakespeare and others.

Non-traditional Authorship Attribution

Non-traditional AA, also referred as automated AA, uses machines to offer a way of capturing authors' writing style, by quantifying some features extracted from internal evidence

[1] See for example shakespeareauthorship.com.

[2] http://www.elizabethanauthors.com/

of documents automatically. Linguists suggest that the internal evidence of writing style consists of two parts: a conscious aspect and an unconscious aspect. The conscious aspect of writing may be controlled and manipulated by authors, while the unconscious aspect of writing is deemed to be independent of authors' will. In this respect, the major hypothesis behind automated AA is that every author has a unique and identifiable style of writing; these characteristics—that usually cannot be consciously manipulated by the author—may plausibly be automatically extracted and reliably measured by computers for style analysis. The terminology *style marker* refers to this type of feature.

Any non-traditional AA approach starts with a corpus of documents in electronic versions that are computer-recognizable. These documents have identified authorship that is believed to be correct. Technically, non-traditional AA is a form of classification on textual data. Therefore, it shares a general framework of text categorization (TC), consisting of two main steps: feature extraction, and classification applied to the extracted features.

The aim of feature extraction is to generate a proper representation for a document. As aforementioned, these features are style markers as they are deemed to be informative of authors' writing habits. However, the extraction of such features is not always straightforward; the fetched electronic versions of documents are not always directly usable for authorship attribution systems. Therefore raw texts are usually pre-processed. Style markers are the key elements in AA; a poor choice of style markers would lead to a severe failure. A wide range of style markers has been proposed in previous research, from lexical-level [Holmes, 1985; Baayen et al., 2002; Diederich et al., 2003; Holmes et al., 2001; Juola and Baayen, 2003] to syntactic-level [Baayen et al., 1996; Kukushkina et al., 2001; Stamatatos et al., 1999]. The lexical-level features can be extracted from surface of the text, such as word length, sentence length, and some vocabulary. In contrast, the syntactic-level features are usually extracted by natural language processing, rather than from the text itself; examples are syntax trees, syntactic annotation, and part-of-speech tags.

Once style markers are determined and extracted, a classification method is then applied to automatically differentiate an author from the others, based on the markers that have been measured. One notable early success was the resolution of disputed authorship in twelve of the Federalist Papers by Mosteller and Wallace [1964]. Many studies since have shown the

Table 1.1: Comparison between traditional AA and non-traditional AA.

	Traditional	Non-traditional
Human effort	Must	N/A
Efficiency	Low	Fast
Size of Corpus	Small	Larger
Collaborative works	Feasible	Not Feasible
Electronic version	Not necessary	Must
Feature types	External and internal	Internal
Features	Limited	Much more
Potential authors	Small workable numbers	Can be much greater
Results	Subjective	Objective

feasibility of implementing automated AA systems. Early studies mainly focused on simple measures, such as Chi-square, usually with small corpora [Efron and Thisted, 1976; Juola, 1997; Juola et al., 2006; Smith, 1983]. In addition, quite a few AA approaches are based on statistical methods, such as cusum techniques [Farringdon, 1996], Markov chains [Khmelev and Tweedie, 2002; Khmelev and Teahan, 2003b], and principal component analysis [Baayen et al., 1996; 2002; Holmes et al., 2001; Burrows, 2002]. More recently, machine learning classifiers have been explored; methods include neural networks [Hoorn et al., 1999; Kjell, 1994a], Bayesian classifiers [Kjell, 1994a; Coyotl-Morales et al., 2006; Uzuner and Katz, 2005], support vector machines (SVMs) [Diederich et al., 2003; Koppel and Schler, 2004], and decision trees [Koppel and Schler, 2003].

Table 1.1 summarises the differences between traditional AA and non-traditional AA. As shown, traditional AA is manual, requiring human effort from experts and scholars. That is, internal evidence within the text itself is examined by experts manually, indicating that traditional AA hardly involves a complicated computation. Moreover, traditional AA relies on external evidence to establish a theory to attribute authorship, meaning that it takes time to make a decision—sometimes several years. In contrast, non-traditional AA makes use of computers, statistical analysis, and machine learning to arrive at an answer immediately. It is concerned with internal evidence only, and the results are more objective. Also, non-

traditional AA systems are mainly designed to attribute authorship to documents that are written by single author, not to collaborative works.

This thesis is concerned with development of effective techniques for non-traditional AA. The abbreviation AA refers to non-traditional authorship attribution in this dissertation, unless explicitly specified otherwise.

1.1.1 Issues Remaining in Authorship Attribution

Effective AA relies on three basic elements. Deep knowledge in linguistics provides good choices of style markers for style analysis. Statistics offers a good way of quantifying these style markers. Classification techniques are of great importance of discriminating the measured style markers correctly. However, although AA has been investigated for the past several decades and over 300 studies have been published, there are many unsolved problems in AA; and, the outcomes are still unsatisfactory. A major indication is the lack of consensus as to methodologies or techniques.

Much previous work in this area is marred by lack of use of shared benchmark data. These collections differ in terms of knowledge domain, size, and genre, which we review in more detail in Chapter 2. Evaluation based on multiple data sets has led to difficulties in comparison between methods. It is almost the case that each paper differs in both style markers and classification method, which makes it difficult to determine which element led to success of the AA approach, or indeed whether AA was successful at all. In most of the published papers, the attribution methods appear to succeed on the terms set by the researchers, but there is no way of identifying which is the most successful. Inconsistencies in the underlying choices also lead to confusion; for example, no two papers use the same sets of style markers. Also, most of the data sets used are small, and changes in performance as documents are added is not examined. It is not clear whether these methods are scalable, reliably effective, or robust.

In more recent years, research in AA has focused on three aspects: designing realistic collections or corpora of texts of known authorship as research testbeds; defining standard attribution tasks for comparing various methodologies and techniques; and standardising evaluation methods. Instead of solving disputed texts in literature, current AA research

tends to concern day-to-day applications.

1.2 Research Contributions

In this thesis we addresses issues in AA from three perspectives: corpora, style markers, and attribution methodologies. Document collections play a critical role in evaluation of any AA system. Results achieved on good data sets are intuitively more reliable than those on poorly designed collections. Our first contribution is that we develop new testbeds in order to evaluate AA techniques. Nine collections of different sizes and kinds are created for different types of AA investigations. Seven of the nine collections are newswire stories in English, and the other two are drawn from English literature. Some of these collections are much larger than those used in prior AA research.

In order to establish the value of using such collections, we undertake a comprehensive comparison of AA methods proposed in previous literature, including naïve Bayesian, Bayesian networks, nearest neighbour, n-nearest neighbour, decision trees, and support vector machines (SVM). We find that AA can be reasonably effective, and a consistent test corpus can be used to distinguish between different approaches to attribution, while it is important to design experiments appropriately. However, the tested machine learning methods are not particularly successful with n-class AA. Results from this preliminary investigation are used as baseline results in this thesis.

One of our primary contributions is that we propose a new methodology for effective and scalable AA, using information theory. The Kullback-Leibler Divergence (KLD) [Manning and Schütze, 1999], or relative entropy, plays the core role of the classifier. This KLD-based framework incorporates language models [Zhai and Lafferty, 2004], borrowing smoothing techniques in information retrieval to estimate probability distributions of the extracted markers. One strong motivation for exploring such an approach is efficiency. The training process is extremely simple, and the computational complexity is almost linear, indicating that our method is more efficient than existing methods, which are typically quadratic or exponential in computational cost. Several smoothing techniques are applied and carefully evaluated, with multiple data collections from different domains. Results are compared to the preliminary investigation; we demonstrate that our method is highly competitive to the

best previous approaches (Bayesian networks and SVMs), and even better than in many previous AA investigations.

This KLD-based method is also a promising alternative to the standard problem of categorization of documents. Reuters newswire data [Lewis et al., 2004], the benchmark data in TC, is used to examine the feasibility and effectiveness of our method being used in another application. We infer that, given appropriate feature extraction methods, the same technique can be used for either problem.

Another contribution in this thesis is that we propose, evaluate, and compare several types of style markers under a consistent experimental setup, from lexical-level features such as function words, to syntactic-level features such as syntax trees. The aim of this investigation is to test which is the best marker type for AA. We find that there is no single type of marker that satisfies all AA applications; each type is superior for some cases but not others; however, function words are generally better than other proposed markers. We extract rich style markers, such as POS tags and syntax tree, by applying natural language processing; however the effectiveness is lower than with simple style markers. We observe that combining multiple types of markers into one feature set does not provide better effectiveness, but worse. Therefore, we present three novel ways to make use of multiple marker types; these approaches can significantly improve AA effectiveness, and each has its own advantages.

We propose a novel task of authorship search (AS). The AS system is, to the best of our knowledge, the first style-based search system. In contrast to conventional information retrieval systems, where the search is concerned with topics or content of the documents, we implement the first style-based search system that is able to search for documents by a certain author. This kind of system is intended for large document collections, in order to address the scalability issue in AA research. Given a query with valid author information, the AS system is able to effectively return documents that are written by the same author as the provided query, within large collections. We carefully evaluate our style-based search system by varying the size of the collections, and the volume of the query text. The largest corpus consists of over half a million news articles, with which the best precision at 10 retrieved documents is around 44%.

We further examine authorship search as a basis for authorship attribution. The moti-

vation of this investigation is to increase the scalability of AA. We show that our method is reasonably effective at discriminating between a few hundred authors with tens of thousands of documents. The results are dramatically better than the existing outcomes in AA, in terms of the number of potential authors, the number of documents, and the difficulty in experimental design.

1.3 Thesis Structure

- **Chapter 2** reviews the state-of-art research in text categorization (TC), as well as authorship attribution (AA). The mathematical background of these state-of-art techniques in both areas is reviewed.

- **Chapter 3** develops suitable data collections for evaluations of AA; nine collections derived from two domains are designed and established for different AA tasks. In this chapter, a preliminary AA investigation is also undertaken on two collections, using state-of-art machine learning techniques that are successful for text classification; the results obtained are used as a baseline throughout this thesis.

- **Chapter 4** proposes a new methodology for effective and scalable authorship attribution (AA). Our method is shown to be superior to the state-of-art methods, including support vector machines and Bayesian networks, in several aspects.

- **Chapter 5** is concerned with various types of style markers that are the key elements in AA. We investigate the effectiveness of these marker types, providing a consistent experimental environment. Based on one of the observations—that is, there is no one type of style marker that can satisfy all AA scenarios—we further propose three systems to significantly improve the AA effectiveness, via different ways of combining evidence.

- **Chapter 6** presents the task of authorship search (AS). This search-based AA is effective and much more scalable than other methods reported in previous AA studies, in particular with large document collections, and with a large number of potential authors. The method is evaluated on several collections; the largest one consists of over half a million documents.

- **Chapter 7** provides a comprehensive comparison between the proposed AA methodologies in this dissertation. Both the classification-based AA and search-based AA are evaluated on a test corpus derived from English literature. Both approaches are shown to be highly accurate.

- **Chapter 8** concludes this thesis and includes a discussion of avenues for future research.

Chapter 2

Background

Authorship attribution (AA) is the task of identifying the author who wrote a particular document. It is usually considered as a type of text categorization (TC) due to the fact that both AA and TC aim to assign documents into a set of pre-defined categories. In TC, documents in a collection are labelled by a large variety of topics; in AA, documents are labelled by potential author candidates.

Both TC and AA share a similar framework for classification of unknown documents, which involves two main steps: feature extraction, and classification that is applied to the extracted features. On one hand, it is worth exploring whether existing TC techniques are applicable for effective AA, and how scalable these approaches are. On the other hand, as we will demonstrated, AA differs from TC in a variety of respects; this fact suggests that a successful TC technique may not guarantee satisfactory results in AA. Therefore, investigating new methods for AA is of greater importance than simply applying existing TC approaches.

In recent years, a variety of AA techniques have been proposed. However in most cases the evaluation of the techniques has been unsatisfactory, and whether the techniques are effective is open to question. For most of the methods it is also not clear whether they can scale beyond trivial problems.

This chapter discusses text categorization (TC), as well as current machine learning methods used for TC. This is followed by an introduction to AA, a contrast between AA and TC, and a discussion of previous AA approaches.

2.1 Text Categorization

Text categorization (TC) is also known as text classification. Techniques are applications that assign documents or texts into one or more pre-defined categories, based on the topics or the content of the documents. The internet boom has led to challenges to information management. Automated TC techniques have been widely investigated for effective information management, that is, to provide users with more effective ways to access the right information. A variety of information management applications have benefited from TC techniques, such as document filtering, document tracking, webpage categorization, and document summarization.

Clustering and classification are two ways to group text data. The general purpose of text clustering and text classification is to divide a collection of texts into different groups, so that documents in the same category share similar properties. In the context of TC, properties usually refer to the topics or the content of documents in the collection. The fundamental difference between text clustering and text classification is whether sample texts are labelled for training purposes. For text clustering, texts are unlabelled, and there is no prior taxonomy available. In this sense, text clustering is also referred to as unsupervised text classification, and the text clustering techniques are considered as unsupervised learning methods. We are not concerned with text clustering in this thesis.

In contrast to text clustering, text classification is based on a learning or training process that uses sample texts. A set of labelled documents are essential for this training process. Initially, models are extracted from the labelled documents for each of the pre-defined topic categories. Then new documents are assigned to one or more categories, based on the extracted models. Therefore, text classification is regarded as supervised classification. The terminologies “classification”, “text classification”, and “text categorization” all refer to supervised text classification throughout this thesis.

Text categorization is the task of assigning texts from a universe to pre-defined categories. Traditional TC was undertaken manually by domain experts. The experts were required to read through each document individually, and then assign the documents to one or more pre-defined categories. The decisions made can be subjective, since they are based on the understanding of certain experts perceived from the documents. Intuitively, traditional TC

is considerably time consuming, and requires a great amount of human effort. This has led to difficulties in dealing with the exponential increase of information over recent years. Automated TC is an attractive alternative to manage access to information. There are a considerable number of techniques that have been previously studied [Bekkerman et al., 2003; Kolcz et al., 2001; Lai and Wu, 2002; Lewis et al., 2004; Li et al., 2003; Sebastiani, 2002; Wolters and Kirsten, 1999; Yang and Pedersen, 1997; Yang and Liu, 1999; Yang, 1999; 2001], a large proportion of which are machine learning (ML) approaches. Relevant background in TC is introduced below, including the definition of TC, a general framework, and discussion of previous research in TC literature.

Given a set of pre-defined categories $C = \{c_1, \ldots, c_i\}$ and a document collection $D = \{d_1, d_2, \ldots, d_j\}$, a TC system is able to assign a Boolean value to each document-category pair $\langle c_j, c_i \rangle$. There are two ways to define TC problems based on the cardinality of the category set $|C|$.

Single-labelled and Multi-labelled. A TC task is single-labelled if only one category c_i is assigned to each document d_j in the collection D. In contrast to single-labelled TC, an overlapping assignment is allowed in multi-labelled TC, that is, the number of categories assigned to each document $d_j \in D$ can be any integer within the range from 1to the total number of categories $|C|$ inclusive, notated as $[1, |C|]$.

Binary and Multi-class. A TC task is binary if the number of pre-defined categories $|C| = 2$, while it is multi-class if $|C| > 2$. Binary TC is rare in practice, nonetheless, it is important due to the fact that a multi-class TC task can be converted to a number of $|C|$ binary TC tasks, where a document d_j is classified into two classes—either c_i or $\overline{c_i}$.

A general TC framework involves two main steps: the extraction of document representations, and a classification methodology that is applied to the extracted representations. For each document-category pair $\langle d_j, c_i \rangle$, a classification function θ is derived that returns either 1 for a correct match or 0 for an incorrect match. Note that, for each document d_j, it is valid to have more than one category $c_i \in C$ that satisfies $\theta(d_j, c_i) = 1$, where θ is a classification function.

2.1.1 Benchmarks in Text Categorization

Text categorization (TC) has been an active research area, and has been widely investigated in recent years [Bekkerman et al., 2003; Sebastiani, 2002]. In TC, researchers have dedicated great effort to achieve consensus on the document collections, attributes shared by the documents, and standard evaluation methods. The proposed methods can be comparable if, and only if, both collections and experimental setup are comparable. There are several corpora that have been developed and freely distributed for research purposes in the area of TC.

Reuters-21578. Reuters collections consist of newswire articles from 1987 to 1991. To date, there have been several versions of Reuters collections that have been experimented with, such as Reuters-22173, Reuters-21450, Reuters-3, and Reuters-21578. Reuters-21587[1] is the the most recent, and has been one of the most popular data sets in TC [Lewis et al., 2004; Joachims, 1998; Masuyama and Nakagawa, 2004; Moschitti and Basili, 2004; Tong and Koller, 2002; Yang, 1999; 2001].

The Reuters-21587 collection was compiled by Lewis et al. [2004] using the documents which were originally collected by the Carnegie group. In contrast to the other versions, some improvements have been made with the Reuters-21578 corpus. First, early versions contained duplicate documents that have been removed. Thus, evaluations of techniques are unlikely to be misled by duplicates. Second, the Reuters-21578 collection includes information with respect to the splitting methods that have been often applied in previous research. The splitting methods indicate how to use documents in experiments; some are chosen for training, some are used for testing, and some are neither used for training nor testing. These specified splitting methods are summarised from prior research in literature, including "Modified Lewis (ModLewis) split", "Modified Apte (ModApte) split", and "Modified Hayes (ModHayes) split". The "ModApte split" has been used the most amongst the three methods.

The Reuters-21578 collection, also known as skewed data, consists of a total of 21,578 documents over 135 categories [Yang and Liu, 1999]. A single document in the collection may have one or more categories; that is, the numbers of categories per document are in

[1] Available from http://www.research.att.com/lewis/reuters21578.html

Table 2.1: The number of documents in the top 8 categories in the Reuters-21578 collection. (The split method used to separate documents is ModApte.)

	Number of documents in the top 8 categories							
group	acq	crude	earn	grain	interest	money	ship	trade
train	1675	2193	1295	2226	2263	2224	2289	2240
test	668	150	1048	117	80	123	54	103

the range 1 to 14 inclusive. From another perspective, some categories contain many more documents than do other. An example is shown in Table 2.1, to demonstrate that documents in the Reuters-21578 collection are not uniformly distributed. The numbers in the table are based on the main category of each document in the collection. We consider the first category entry that appears in the tag component "<TOPICS><D> $\cdots$ < /D>< /TOPICS>" of any document as the main category for that particular document. Finally, 82% of the categories consist of less than 100 training documents; 33% of the categories have less than 10 training samples.

Ohsumed-233445. This is a collection of bibliographical documents that were originally compiled by Hersh and colleagues at the Oregon Health Sciences University [Hersh et al., 1994].[2] It is a subset of documents from the MEDLINE database.

Originally, there were 348,566 reference entries that were collected from 270 medical journals between 1987 and 1991. Only 233,445 of the entries, provided with abstracts as well as titles, were chosen to form the Ohsumed-233445 corpus.

Ohsumed-2334425 data, another collection that has been used in TC [Joachims, 1998; Moschitti and Basili, 2004; Yang, 1999; 2001], covers 180 categories.

20NG. This collection consists of 19,997 articles of 20 different categories in total, in which texts are taken from Unsenet newsgroups.[3] Although 20NG is less popular than the previous two collections, it has been investigated in several studies [Moschitti and Basili, 2004; McCallum and Nigam, 1998a].

[2]Available from ftp://medir.ohsu.edu/pub/ohsumed

[3]http://www.ai.mit.edu/people/jrennie/20Newsgroups/

Others. There are other collections that are not as popular. For instance, Yang [2001] has experimented with HV-28 data and HV-255 data. Both collections consist of the same set of 4,285 synthetic webpages. Another example is the TREC9-MeSH-Bath data that has been used by Yang et al. [2003]. It is a TREC-9 version of the Ohsumed data, from the subset of "MeSH" category.

2.1.2 Document Representation

Once a document collection is chosen, extraction of document representations is then the first technical problem to be investigated. The unit components in a document representation are called features. The extraction process starts with text pre-processing, in which junk information such as disclaimers from certain publishers and mark-up tags is removed. The commonly used text pre-processing methods in TC include case-folding, stopping, and stemming. These approaches are similar to those used for indexing purposes in information retrieval (IR).

Case-folding is the process of transforming all alphabetic characters to the same case. Case-folding is occasionally problematic, when for example the word is actually an acronym for instance.

Stopping deals with removing semantic-free but grammatically important words [Witten et al., 1999]. A closed-class list of words is usually pre-defined. These words are considered to be independent of topics or content, so that they are of little importance in TC. However, it is not trivial to decide which terms should be collected into the stoplist, and there is no consensus. Different TC systems usually have different lists of words for stopping.

Due to the observation that words often have several morphological forms that share similar semantic meanings, stemming is suggested to remove common prefixes and suffixes from words. For example, "difficulties" and "difficulty" can be stemmed to the word root "difficult". It is understood that stemming is able to reduce the feature space greatly, and is also able to boost the efficiency of TC systems. On the other hand, stemming may introduce ambiguity that misleads classification. For example, both "university" and "universe" can be stemmed to the same root, that is "univers"; the original words have completely different semantic meanings, but become identical in the final document representation after process-

ing. Choice of stemming algorithms is important; several algorithms have shown the power of improving the effectiveness of both TC and IR systems [Frakes, 1992; Frakes and Fox, 2003; Porter, 1980].

After the three steps of text pre-processing, features are extracted to form the final representations of documents. These representations are directly input to a TC system. Given a set of pre-defined categories, features have certain discrimination power in relation to each of the topic categories. Most TC approaches treat documents as bags-of-words; each feature is, in fact, a word or word root after text pre-processing [Yang and Liu, 1999; Yang, 1999]. However one major problem of text categorization is the high dimensionality of the feature terms in the feature space are relevant for classification. Therefore, it is straightforward to use feature selection techniques to reduce dimensionality of the feature space [Gabrilovich and Markovitch, 2004; Lai and Wu, 2002; Kolcz et al., 2001; Shang et al., 2007].

An alternative approach is to propose new features other than bags-of-words, such as phrases [Fuhr and Buckley, 1991; Tzeras and Hartmann, 1993; Schütze et al., 1995], word sequences [Mladenic and Grobelnik, 1998; Lewis, 1992b; Dumais et al., 1998], distributional clusters of words [Bekkerman et al., 2003], character-level n-grams [Li et al., 2006], and substrings extracted from texts [Zhang and Lee, 2006]. Moreover, shallow natural language processing (NLP) has been proposed for TC in recent years. Masuyama and Nakagawa [2004] proposed the use of part-of-speech (POS) tags for a feature selection stage, which the effectiveness was improved. However, NLP is not always helpful; for example, Dumais et al. [1998] applied NLP, but were unable to achieve a improvement in performance.

2.1.3 Feature Weighting in Text Categorization

Any TC system represents documents as sequences of features that are to be classified; as an input to a classification function, these features are weighted with values that are usually normalised within the range of [0, 1]. For some special cases, researchers may apply a binary weighting scheme: values of features extracted from documents are valued either 0 for absence, or 1 for presence [Schütze et al., 1995; Sebastiani et al., 2000]; such weighting is not common in TC.

The most widely used weighting scheme in TC is an IR-style weighting—the standard $tf \cdot idf$, which is based on the vector space model [Salton and Buckley, 1988]:

$$tf \cdot idf \quad = \quad tf\left(f_k, d_j\right) \cdot \log_2 \frac{N}{df\left(f_k\right)} \tag{2.1}$$

where tf represents term-frequency; $tf\left(f_k, d_j\right)$ is the number of times that the feature f_k occurs in the document d_j; $df\left(f_k\right)$is the number of documents in which the feature f_k occurs, and N refers to as the number of documents available for training for certain categories. The aspiration of idf—the inverse document frequency—is that terms or features that occurs in many documents in the collection have be less discriminative power. To assure that the values of $tf \cdot idf$ fall within the range of 0 to 1, normalization such as by the length of the document is usually applied. This weighting scheme originated from IR, and any variants of calculation of $tf\left(f_k, d_j\right)$ and $df\left(f_k\right)$ proposed in IR literature may be used for TC; some detailed discussions about these variants were provided by Zobel and Moffat [1998], and Singhal et al. [1996].

Although $tf \cdot idf$ is so far the most popular technique for term weighting, other schemes were also proposed, such as using probabilistic approaches [Gövert et al., 1999; Fuhr and Buckley, 1991]. Schemes proposed for feature selection in recent years include for example, DIA association factor [Fuhr and Buckley, 1991], information gain [Caropreso et al., 2001], χ^2-statistics [Yang and Pedersen, 1997; Yang and Liu, 1999; Sebastiani et al., 2000], mutual information (MI) [Dumais et al., 1998; Larkey and Croft, 1996; Lewis and Ringuette, 1994]. These aim to reduce the high dimensionality in the feature space, while achieving the highest effectiveness. However, these techniques are beyond our scope of this research, and therefore we do not present detailed reviews on these topics in this thesis. More discussion can be obtained from Sebastiani [2002].

2.2 Machine Learning for Text Categorization

Machine learning (ML) is a broad subfield of artificial intelligence. It is concerned with the development of learning algorithms that build models from sample observations. Models are then used to improve the system effectiveness by observing examples, accumulating successful cases, and minimizing failures. Many machine learning algorithms use statistical methods in the process of learning.

Machine learning has been used in a wide spectrum of applications, such as fraudulent transaction detection, classifier construction for pattern recognition, and text categorization. There are two types of learning: supervised learning and unsupervised learning. Broadly speaking, a supervised learning method generates a function that maps inputs to outputs. One standard formulation of supervised learning is the classification problem. The learner is required to learn or to approximate the behavior of a function that maps an input instance into an output, that is, one of several pre-defined classes, by referring to a set of input-output examples available of that function. In the research field of TC, many state-of-art results are obtained by applying machine learning approaches that are supervised learning [Sebastiani, 2002]. In all ML approaches used in TC, a model is learned for each of the categories $c_i \in C$ during the learning phase. Note that learning relies on the observations of a set of instances. The constructed models are then used to predict the likely outputs corresponding to new inputs, as well as instances that have not been seen in the learning process. Many supervised learning classifiers have been successfully proposed for effective TC, including naïve Bayesian, Bayesian networks, k-nearest-neighbour, decision trees, and support vector machines (SVMs). The technical background of these methods is introduced later in this chapter.

In contrast to supervised learning, unsupervised learning methods model a set of inputs without explicit labels. In other words, there are no prior input-output observations available in this kind of learning. Other types of learning, such as semi-supervised learning, reinforcement learning, and transduction [Witten and Frank, 2000] have also been widely investigated. However, they have rarely been used in TC, and we do not examine these techniques in this thesis.

2.2.1 Data Overfitting

Overfitting is an open challenge in ML. A learning algorithm is trained on sets of sample data in order to make predictions on new data. The aim of a learning process is to maximize the predictive effectiveness on new data rather than that on sample data. It is often the case that the best fit to training data may contain some noise that is caused by, for instance, memorizing the peculiarities in the training data, and this is not satisfactory for making predictions on new data. This phenomenon is known as data overfitting. Overfitting can

have a number of causes: the training data size is too small, the amount of noise is too much, and the function learned from data is too complex. Overfitting can either over-estimate or under-estimate the effectiveness of learning approaches.

The learner is assumed to reach a state where enough instances have been observed, and then the learned model is able to effectively predict the output for other examples, including examples that are not presented during training. However in cases such as the learning process being too long, the training examples being rare, or the numbers being too small, the learner may skew to very specific features of the training data. The consequence is that the learned model may have little relation to the target function, and the effectiveness of the model on the training examples may increase while the effectiveness on the unseen data may decrease.

2.2.2 Avoiding Overfitting

Several validation methods have been suggested for evaluations to avoid data overfitting: train-and-test, hold-out cross validation, n-fold cross validation, leave-one-out cross validation, and stratified cross validation. All of them are concerned with how collections can be used. The choice of evaluations usually depends on the amount of data that are involved.

Train-and-Test

Given a set of data, a proportion of the data is selected at random, usually about 20% to 30%, and is used as the test data. The remaining data is used to train the classifier that is evaluated on the reserved test data. Generally, this approach is suitable for data collections of reasonably large size. However, selections of test data are indeed task dependent. For instance, if there are millions of samples available, it is not wise to use the majority of the data for training. Alternatively, 10% of the data may be selected for learning, and the remaining data can be used for testing. This is because, first, too much training data can easily cause overfitting. Second, the learning process on a large amount of data is very expensive and inefficient.

Cross validation

For collections of moderate size, the commonly used method is cross validation. This involves grouping data into m subsets. The samples in each subset are predicted by the classifiers trained from the samples in the remaining $m-1$ subsets. Eventually, a total of m classifiers are learned. Estimation of the classification technique is carried out by averaging the effectiveness of the m classifiers. There are several variants of cross validation based on different ways of constructing the subsets: hold-out cross validation, n-fold cross validation, leave-one-out cross validation, and stratified cross validation.

In hold-out cross validation, the data is partitioned into two disjoint parts, one for training and the other for testing. Learning is on input-output pairs that are observed from the training set; the true error estimation is achieved by classifying the instances in the testing set. A typical split is to reserve 2/3 of the data for training and 1/3 for testing.

For n-fold cross validation, data is randomly split n times. The union of all test splits forms the full data set. The classifier predicts for instances in each of the test splits, by learning from the remaining data—that is, $n-1$ sets of data. The effectiveness of the classifier is averaged from the estimations of all the n splits. Ten-fold cross-validation is a standard instance of n-fold cross-validation.

Leave-one-out cross validation is regarded as a special case of n-fold cross validation, in which n is the number of total instances in the data set. Each instance is left out and predicted in turn. The effectiveness of a classifier is estimated by predicting all instances.

Stratified cross validation is another variant of n-fold cross validation, where the distribution of training and testing samples in each of the split subsets should be the same as in the original data set.

2.2.3 Evaluations in Text Categorization

Widely used evaluation metrics in TC are: classification accuracy, classification error rate, precision, recall, precision-recall-breakeven point, F-measure, and averaging F-measure. All of these measures are concerned with the effectiveness of a TC system.

Given a set of documents to be categorised, decisions made by any TC system can be grouped into four different types, as shown in Table 2.2. Different evaluation metrics are

Table 2.2: The confusion matrix for category $c_i \in C$. $\overline{c_i}$ refers to categories other than c_i.

	c_i	$\overline{c_i}$
c_i	TP_i (true positives)	FP_i (false positives)
$\overline{c_i}$	FN_i (false negatives)	TN_i (true negatives)

derived by using these different kinds of values in various ways. Recall Section 2.1, where the notation c_i indicates a category in a set of pre-defined categories C, d_j refers to a document, and θ is a classification function for each document-category pair $\langle d_j, c_i \rangle$. The notations introduced in Table 2.2 are defined as follows:

- TP_i, true positives, the number of documents correctly classified as being in c_i
  $TP_i = |\,(\theta\,(d_j, c_i) = 1) \cap (d_j \in c_i)\,|$
- TN_i, true negatives, the number of documents correctly classified as not belonging to c_i, that is $\overline{c_i}$
  $TP_i = |\,(\theta\,(d_j, \overline{c_i}) = 1) \cap (d_j \in \overline{c_i})\,|$
- FP_i, false positives, the number of documents incorrectly classified in c_i
  $FP_i = |\,(\theta\,(d_j, c_i) = 1) \cap (d_j \in \overline{c_i})\,|$
- FN_i, false negatives, the number of documents incorrectly classified in $\overline{c_i}$
  $FN_i = |\,(\theta\,(d_j, \overline{c_i}) = 1) \cap (d_j \in c_i)\,|$

Classification Accuracy and Classification Error Rate

Classification accuracy (Acc) gives the percentage of correct decisions made by a classifier. The error rate (Err), as its counterpart, gives the percentage of incorrect decisions:

$$Acc = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{2.2}$$

$$Err = \frac{FP_i + FN_i}{TP_i + TN_i + FP_i + FN_i} \tag{2.3}$$

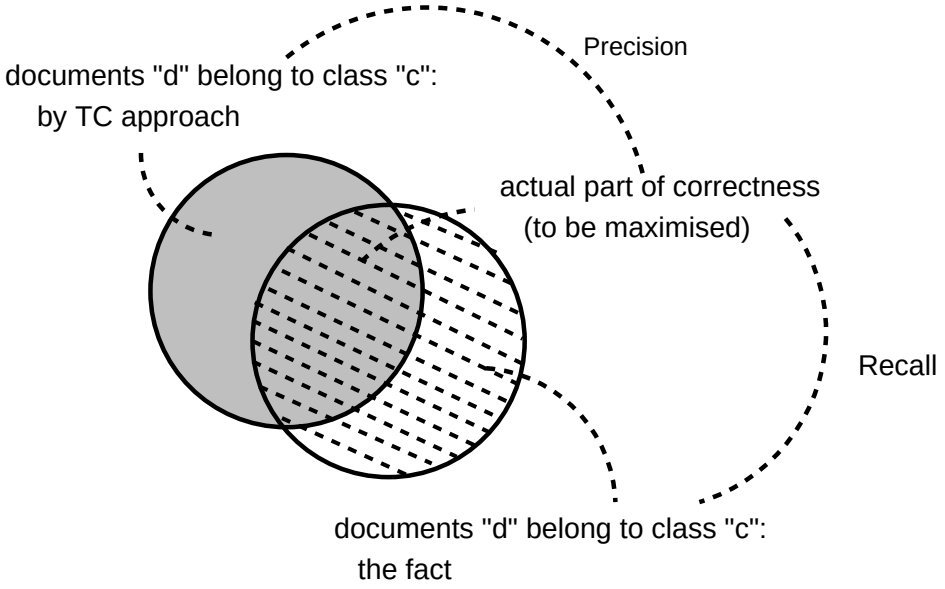


Figure 2.1: Relationship between precision and recall.

Precision, Recall, and Precision-Recall-Breakeven Point

Precision and recall are standard measurements used to evaluate search engines in the field of IR; in recent years, they have been widely used to evaluate TC systems. Precision (Pr) is the percentage of correctly classified documents out of all documents that are classified to the category $c_i \in C$. Recall (Re) is the percentage of correctly classified documents out of all documents in the category $c_i \in C$. The relationship between precision and recall is depicted in Figure 2.1.

$$Pr = \frac{TP_i}{TP_i + FP_i} \tag{2.4}$$

$$Re = \frac{TP_i}{TP_i + FN_i} \tag{2.5}$$

The precision-recall-breakeven point is a point at which the precision equals recall [Lewis, 1992b]. However, as discussed by Sebastiani [2002], precision rarely equals recall in most cases in TC. Thus, an alternative point at which the smallest difference is achieved between precision and recall is then used.

Other evaluation metrics balance values of precision and recall. Two popular evaluations

are F-measure and averaging F-measure. The F-measure is defined as:

$$F_\beta(Pr, Re) = \frac{\left(\beta^2 + 1\right) Pr \cdot Re}{\beta^2(Pr + Re)} \tag{2.6}$$

where the parameter β is used to adjust the input weight from Pr and Re. The F_1-measure is obtained when recall and precision are weighted equally, with $\beta = 1$.

Microaverage & Macroaverage

Data collections used in TC are usually skewed, that is, some categories my have large numbers of documents for both training and testing while others have little. In this case, the effectiveness of a TC technique can vary dramatically when tested with documents from different categories. To estimate the unbiased effectiveness of a TC technique across all categories, averaging is applied. There are two ways to carry out the "averaging": microaverage and macroaverage. For macroaverage, precision-recall values over all categories are averaged:

$$\overline{Pr} = \frac{\sum_i \frac{TP_i}{TP_i + FP_i}}{|C|} \tag{2.7}$$

$$\overline{Re} = \frac{\sum_i \frac{TP_i}{TP_i + FN_i}}{|C|} \tag{2.8}$$

Macroaveraging simply averages the values across all categories, providing an equal weight to the performance of each category. Microaveraging averages value across all documents, providing an equal weight to the performance of each document. Then the averaged precision/recall is calculated by summing individual unit values as shown in Table 2.2.

$$\overline{Pr} = \frac{\sum_i TP_i}{\sum_i (TP_i + FP_i)} \tag{2.9}$$

$$\overline{Re} = \frac{\sum_i TP_i}{\sum_i (TP_i + FN_i)} \tag{2.10}$$

The microaverage precision/recall and macroaverage precision/recall can be used to derive the microaverage/macroaverage F-measure based on Equation 2.6. Also, these measures can be used to evaluate the effectiveness of a binary classifier over multiple categories.

2.2.4 Text Classifiers

Dozens of machine learning approaches have been proposed for a wide range of applications. In the following sections, we focus on six well-known machine learning classifiers that are

reported to be competitive in TC in terms of effectiveness: naïve Bayesian, Bayesian networks, nearest-neighbour, k-nearest-neighbour (k-NN), decision trees, and support vector machines (SVMs).

Naïve Bayesian Classifiers

Naïve Bayesian classifiers are competitive in practice for both TC and IR tasks [Joachims, 1998; Larkey and Croft, 1996; Lewis, 1992b;a; 1998; McCallum and Nigam, 1998b; Yang and Liu, 1999; Chakrabarti et al., 1997; Frietag and McCallum, 1999]. They are simple probabilistic classifiers based on Bayesian probability theory. It is assumed that the occurrences of features are mutually independent in use of a naïve Bayesian classifier.

Given a pre-defined set of categories C=$\{c_i|i = 1, \ldots, m\}$ and a document d, the task is to calculate the probability of document d belonging to a particular category c_i. The document d can be represented as a set of features $\{f_i|i = 1, \ldots, n\}$. According to Bayesian theorem the statement can be expressed as:

$$P\left(c_i|f_1, \ldots, f_n\right) = \frac{P(c_i) \cdot P\left(f_1, \ldots, f_n|c_i\right)}{P\left(f_1, \ldots, f_n\right)} \tag{2.11}$$

As shown in Equation 2.11, the value of expression $P\left(f_1, \ldots, f_n\right)$ does not rely on any specific category and therefore the denominator of the fraction is effectively a constant across all instances. Removing this component does not affect the relative relationship amongst all $P\left(c_i|f_1, \ldots, f_n\right)$. Therefore Equation 2.11 can be simplified as:

$$P\left(c_i|f_1, \ldots, f_n\right) \propto P(c_i) \cdot P\left(f_1, \ldots, f_n|c_i\right) \tag{2.12}$$

By applying the chain rule, we can reformulate the joint probability on the right hand side of Equation 2.12:

$$\begin{aligned} P(c_i) \cdot P\left(f_1, \ldots, f_n|c_i\right) &= P(c_i) \cdot P(f_1|c_i) \cdot P(f_2, \ldots, f_n|c_i, f_1) \\ &= P(c_i) \cdot P(f_1|c_i) \cdot P(f_2|c_i, f_1) \cdot P(f_3, \ldots, f_n|c_i, f_1, f_2) \\ &= P(c_i) \cdot P(f_1|c_i) \cdot P(f_2|c_i, f_1) \ldots P(f_n|c_i, f_1, \ldots, f_{n-1}) \end{aligned}$$

To simplify the computation of conditional probabilities shown in the above formula, an assumption is made that the occurrence of each feature f_i is independent from its precedent

features. Therefore, the formula can be derived as:

$$P\left(c_i|f_1,\ldots,f_n\right) \propto P(c_i)\cdot \Pi_i\, P\left(f_i|c_i\right) \tag{2.13}$$

$$c_i = \text{argmax}_{c_i\in C}\, P\left(c_i\right)\cdot \Pi_i\, P\left(f_i|c_i\right) \tag{2.14}$$

$P\left(c_i\right)$ is a prior probability, which can be estimated from the available sample set by, for example, measuring the frequency with which a category c_i occurs in the training data, or by simply making it a constant. Although the independence assumption violates many real circumstances, it works effectively in practice [Domingos and Pazzani, 1997; Wang and Zhang, 2005].

In addition, it is problematic to estimate the conditional probabilities $P\left(f_i|c_i\right)$ if sample data is limited. This is due to insufficient occurrences, or even absence, of features in the training data. In this case the observed probabilities of rare features may be too specific to be representative of a category. Moreover, unseen instances cause zero probabilities that can mislead the classifiers.

There are several ways to estimate the probabilities of features that are missing in the training samples. Statistical distributions, such as Gaussian or Poisson, can be used to approximate the distributions of features in the training data. For instance the Gaussian distribution can be given as:

$$\begin{aligned} P(f_i|c_i) &= g(f_i,\mu_i,\sigma_i) \quad \text{where} \\ g(f,\mu,\sigma) &= \frac{1}{\sqrt{2\pi\sigma}}\, e^{\frac{(f-\mu)^2}{2\sigma^2}} \end{aligned}$$

where μ_i is the mean value of feature f_i in category c_i and σ_i is the standard deviation of feature f_i in category c_i. Other alternatives have also been proposed for estimation of parameters in naïve Bayesian classifiers. However, parameter estimation is not our focus, and detailed discussion is omitted in this thesis.

Bayesian Network Classifiers

As the independence assumption, made for naïve Bayesian classifiers, is often misleading, researchers have investigated ways to improve the effectiveness of naïve Bayesian classifiers, by relaxing the independence assumption. Bayesian networks are classifiers of this type,

in which no strict independence assumption is required [Buntine, 1991; 1996; Cooper and Herskovits, 1992; Friedman and Goldszmidt, 1996a;b; 1998; Heckerman et al., 1995].

Bayesian networks are directed acyclic graphs (DAG), which provide representations of the joint probability distributions over a set of random variables. The Bayesian networks relax the independence assumption in that each random variable is independent of its non-descendants in the graph, given its parents. Each node in the graph represents a feature extracted from a document and each edge represents the dependencies between two features or two variables. It is valid for a node to have more than one parent, or no parent. Each node has a table of transition probabilities, in the form of conditional probabilities of the node given its parents.

There are two learning steps in the use of a Bayesian network classifier: learning of network structure, and learning of conditional probability tables for each node in a network. The objective of learning a Bayesian network is to generate a network that can best describe the dependent probabilities over the training data. The network structure is determined by identifying features with the strongest dependencies between them. These dependencies in a Bayesian network are known as conditional dependencies.

Consider a finite set of discrete random variables, $F = \{f_1, f_2 \dots f_n\}$, where each f_i represents a feature extracted from a document. In the context of a Bayesian network, an acyclic annotated graph, each vertex in the graph corresponds to one of the features f_i. Each edge represents the dependency between the two features that are connected by that edge. The network is then interpreted as meaning that each feature f_i is independent of its nondescendants, given its parents. In addition, each feature f_i has a posterior probability distribution derived from its parents.

Classification involves the computation of the joint probability of features $f_1, \dots, f_n$, taking dependencies into account. The probability of a feature f_i is unconditional if f_i has no parents, otherwise it is conditional. For each feature f_i, more than one parent, notated as π_{f_i}, can be involved in evaluating the joint probability. Therefore, the joint probability of a set of features can be derived as:

$$P(f_1, \dots, f_n) = \Pi_i P\left(f_i | \pi_{f_i}\right) \tag{2.15}$$

To carry out the numerical calculations, we need to specify for each node f_i the probability

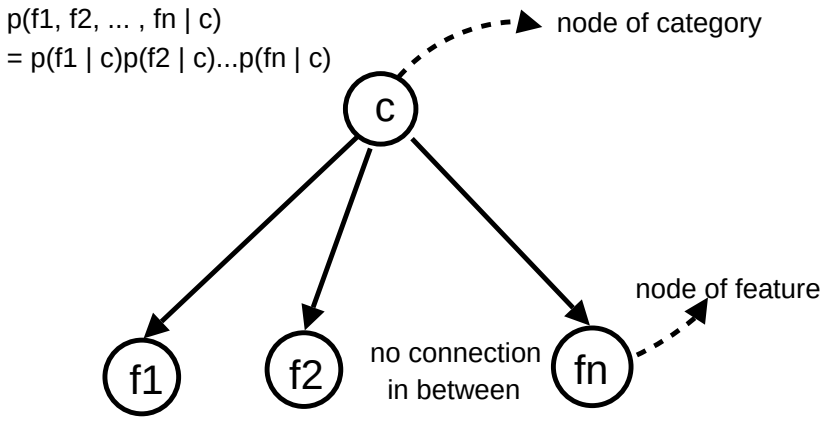


Figure 2.2: Network structure of a naïve Bayesian classifier.

distribution for f_i conditional on its parent notes, notated as π_{f_i}. There are several ways to estimate the conditional probabilities $P(f_i|\pi_{f_i})$. However for simplicity, it is common to deal with discrete or Gaussian distributions.

A naïve Bayesian classifier can be represented as a Bayesian network of the simplest structure as depicted in Figure 2.2. Node c is a classification node that indicates one of the categories. In the context of text categorization, f_1 to f_n are features that can be extracted from documents in the collection. Based on the independence assumption, the only connections in this network are between the classification node and each of the features. No other connections are allowed.

Bayesian networks relax the independence assumption made in naïve Bayesian classifiers. Categories are represented as one of the feature nodes in a Bayesian network. In this sense, it can be considered as an unsupervised learning method as the learner does not distinguish the category variables from the feature variables.

A Bayesian network is able to handle missing values in the training data, and prediction on new data can be made by inference in that Bayesian network, given the network structure as well as the tables of conditional probabilities in associated with each vertex in the network. The advantage of a Bayesian network classifier over a naïve Bayesian classifier is that the missing values can be approximated from a set of observed instances other than from the pre-defined distributions. The example below illustrates how to use Bayesian networks to

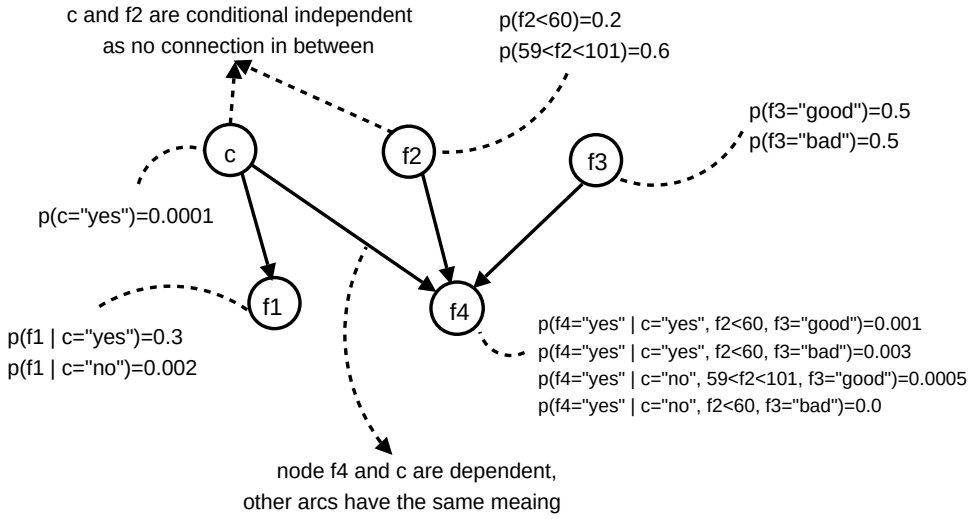


Figure 2.3: A simple Bayesian network. Here, c is the category node and each f_i represents a feature. The arcs indicate the dependencies between features. Each node holds a a table of conditional probabilities.

handle missing values.

Given a simple Bayesian network as depicted in Figure 2.3, the nodes are either features or categories extracted from training data, and the arcs indicate the dependencies between the connected nodes. Nodes with no arcs in between are conditionally independent, of which the dependencies can be neglected in the calculation process. The probabilities defined in Equation 2.15 are indicated by the arcs in the network.

The ordering of features is important in determining the network structure. For example, using the ordering (c, f_2, f_3, f_1, f_4) generates the following conditional independencies and the obtained structure is as shown in Figure 2.3:

$$p(f_2|c) = p(f_2) \tag{2.16}$$

$$p(f_3|c, f_2) = p(f_3) \tag{2.17}$$

$$p(f_1|c, f_2, f_3) = p(f_1|c) \tag{2.18}$$

$$p(f_4|c, f_2, f_3, f_1) = p(f_4|c, f_2, f_3) \tag{2.19}$$

However, if a different ordering of features is chosen, the network structure can differ

greatly. When the structure of a network is complex and the number of cliques[4] in the network is large, the computational complexity of a Bayesian network is high, increasing exponentially with the number of training documents and the number of cliques. The worst case in our example is to have a fully connected network. In this case the computational complexity will be $n!$, given that each feature node f_i has n parents.

Alternatively, instead of pre-choosing a appropriate ordering of features, observations of causal relationships from the conditional dependencies can be used to determine the structure of the network.[5]

Once a Bayesian network is constructed, the final step is to calculate the probability distributions of $p(f_i|\pi_i)$ for each feature f_i. However, it is often the case that some of the probabilities are not directly stored in the network. For example, the probability of c given observations of the other features, that is $p(c|f_2, f_3, f_1, f_4)$, is not directly observable from the Bayesian network as shown in Figure 2.3. But it can be inferred as follows:

$$p(c|f_2, f_3, f_1, f_4) = \frac{p(c, f_2, f_3, f_1, f_4)}{p(f_2, f_3, f_1, f_4)} \tag{2.20}$$

By applying the chain rules to the equation above, it is possible to use local probabilities and the prior knowledge of conditional independence to calculate the probabilities:

$$p(c, f_2, f_3, f_1, f_4) = p(c)\, p(f_2|c)\, p(f_3|c, f_2)\, p(f_1|c, f_2, f_3)\, p(f_4|c, f_2, f_3, f_1) \tag{2.21}$$

The above equation can be simplified by using the equivalent replacement derived from Equations 2.17 to 2.19:

$$p(c, f_2, f_3, f_1, f_4) = p(c)\, p(f_2)\, p(f_3)\, p(f_1|c)\, p(f_4|c, f_2, f_3) \tag{2.22}$$

The denominator $p(f_2, f_3, f_1, f_4)$ in Equation 2.20 can be derived in a similar way. All the conditional probabilities on the right hand side of the equations can be directly found from the local tables of nodes as shown in Figure 2.3. The values associated with each node are the probabilities of that node conditioned on its parent or parents. Bayesian networks

[4] A clique is a subset of nodes, which is complete, meaning that there is an edge between every pair of nodes in this subset; the minimum number of nodes in the set is two.

[5] In use of this approach, observations that may be relevant to the problem have to be identified beforehand. It is usually not unique to modelling with Bayesian networks, and there are no explicit solutions.

are not always ideal methods for applications involving a large number of data and complex relationships due to the high computational cost.

We have described the fundamental theory of how to use a Bayesian network for prediction. However, our description is far from complete; many studies have been published from various perspectives to improve Bayesian networks, such as learning probabilities in Bayesian networks, using Bayesian networks for continuous data, and reducing the computational cost of Bayesian networks. These topics are beyond our scope; a comprehensive tutorial on Bayesian networks is provided by Heckerman [1995, Revised 1996].

Nearest Neighbour & K-Nearest Neighbour Classifiers

K-nearest-neighbour is an instance-based (IB) learning algorithm. Instance-based (IB) learning methods store the training examples and postpone the learning process until a new instance must be classified. Such learning methods are also known as lazy learning, due to the deferred process of learning. The instance-based classifiers have been successfully used for pattern classification on many applications [Cover and Hart, 1967].

A k-nearest-neighbour (k-NN) classifier assumes that each of the instances in the data corresponds to a particular point in an n-dimensional space. In the context of English texts, each document containing n features is represented by a data point in an n-dimensional space. The distance D between two points or two documents, d_i and d_j for instance, is usually computed by a p-norm distance function:

$$D_{d_i,d_j} = \sqrt{p}\sum_{k=1}^{k=n}(|d_{ik} - d_{jk}|)^p \tag{2.23}$$

where k indicates the features in documents d_i and d_j. The value of p represents the norm of the distance calculated. In principle, data points that are close are regarded as in one cluster, and therefore, the documents associated with these points are then classified as in the same category. The p-norm distance is easy to compute, and is effective. Standard Euclidean distance has been widely used, where $p = 2$.

In text classification, a k-nearest-neighbour algorithm (k-NN) aims to classify texts or documents based on the closest training examples in the n-dimensional feature space. The category of a document is determined by a majority vote of its metrically nearest k neigh-

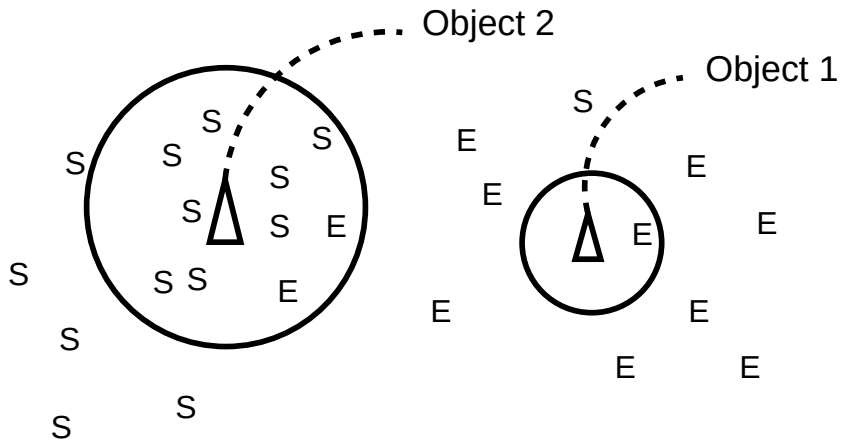


Figure 2.4: An example of a k-NN classifier.

bours. The nearest-neighbour algorithm is the simplest variant of k-NN algorithm, where $k = 1$. The category of the closest sample to an unknown text is assigned as the expected class of that text.

An example of a k-nearest-neighbour classifier is shown in Figure 2.4. Symbol E represents the topic education and S for the topic sports. The two triangle objects in the figure are the new instances to be classified, and the classification is made to either of these two topic categories. As shown, the nearest symbol to Object 1 is E, therefore, it is attributed to the education category by a simple nearest-neighbour classifier. For Object 2 on the other hand, if we are interested in ten nearest-neighbours of the object then a k-NN classifier is adopted, given $k = 10$. It is observed that the S symbols occupy 80% of the data points nearby, therefore Object 2 is classified to be in the sports category. The number of nearest-neighbours k is a parameter that can be adjusted. The best choice of k depends upon the data itself. In general, larger values of k are preferred, which is believed to be able to effectively reduce the data overfitting caused by noise. However, larger k values also make the boundaries between classes less distinct.

As discussed by Aha et al. [1991], these kinds of methods make no assumption of probability distributions of features, and the decision boundaries between classes constructed by k-NN can be very complex. In many previous TC studies, the k-NN classifier has been

considered as one of the best method for TC. Yang [1999] compared 14 TC methods in effectiveness; several test collections were used, including different versions of Reuters newswire data sets and Ohsumed data sets. The results showed that the effectiveness of a TC classifier is task dependent: it varies dramatically when using different data sets for evaluation. Even multiple versions of the same data cannot guarantee a similar result. Amongst all the methods, only the simple k-NN classifier has been able to scale up to the entire Ohsumed data. Unlike other learning methods, k-NN classifiers do not require any prior training or learning process. The classification is made by measuring the distances between pairs of documents consisting of the new example and one of the training examples. The drawback is that, for each instance to be classified, the calculations of distances have to be carried out over all instances in the entire collection, meaning that the efficiency of using such a method is an issue, and less plausible for collections of massive data.

Decision Tree Classifiers

Decision trees are simple but competitive inductive learning methods that have been successfully used in TC [Li and Yamanishi, 1999; Yang, 1999; Gabrilovich and Markovitch, 2004]. The basic idea is straightforward. Given training samples and a set of pre-defined features that can be extracted from samples, a tree is constructed to describe the relationship amongst features. Each feature is represented by a non-leaf node in the constructed tree, and the categories are represented by leaf nodes or external nodes in the tree. The internal or non-leaf nodes are the decision points where features can be differentiated. The class of an instance is determined by the leaf node that the instance reaches when it traverses along the tree.

A critical issue in constructing a decision tree is to select appropriate features at particular splitting points. This directly affects the classification effectiveness. A general method for selection of features is to compute the information gain for each feature. The information gain measures how well a given attribute separates the training examples according to their target classification. This measure is used to select among the candidate attributes at each step while growing the tree. The hypothesis of this approach is that the higher information gain a feature has, the better it separates the data. The information gain is derived from entropy; given a data set S containing samples of a total of i classes, the entropy of such

data can be measured:

$$Entropy\,(S) \quad = \quad -\sum_{i} p_i \log_2 p_i \tag{2.24}$$

where p_i is the probability of samples in the class i. Entropy measures the uncertainty of the training data, and information gain (IG) is known as the expected reduction in entropy, caused by partitioning the examples according to a certain attribute. The information gain of a feature j can be measured by:

$$IG\,(S, j) \quad = \quad Entropy\,(S) - \sum_{v \in Value(j)} p_v \; Entropy\,(S_v) \tag{2.25}$$

$$\propto \quad - \sum_{v \in Value(j)} p_v \; Entropy\,(S_v) \tag{2.26}$$

where $Values\,(j)$ is the set of all possible values for attribute j, and p_v is the probability of a sample for which the feature j has value v. Note that, the first part in the equation is the entropy of the original data S, and the second part is the expected value of the entropy after S is partitioned using j. In other words, the expected entropy described by this second part is the sum of the entropies of each subset S_v (containing samples for which the value of feature j is v), weighted by the probability of p_v. The same process is applied recursively to different features to generate branches of the tree; the feature j with the largest information gain is selected as the root of the tree. The information gain has been used in several successful decision tree algorithms, such as C4.5, C5.0, ID3, and J48 [Quinlan, 1993]. Amongst these variants, C4.5 is the most popular algorithm in the decision tree family.

The construction of a decision tree is a process of learning a decision plan, which takes a set of attributes as the input and arrives at a decision class as the output. When a new instance—represented by a set of attributes—is supplied to a decision tree, it traverses the tree until a leaf node is reached. The class associated with the leaf node is then selected as the class of the supplied instance. Paths of a tree are usually kept short, as each feature occurs only once as a node at a particular splitting point. In this sense, only a subset of features are considered when classifying the given instances.

A simple example of a decision tree is shown in Figure 2.5. The leaf nodes are labelled by classes c_1 and c_2, and other nodes are features extracted from the sample data. Only three

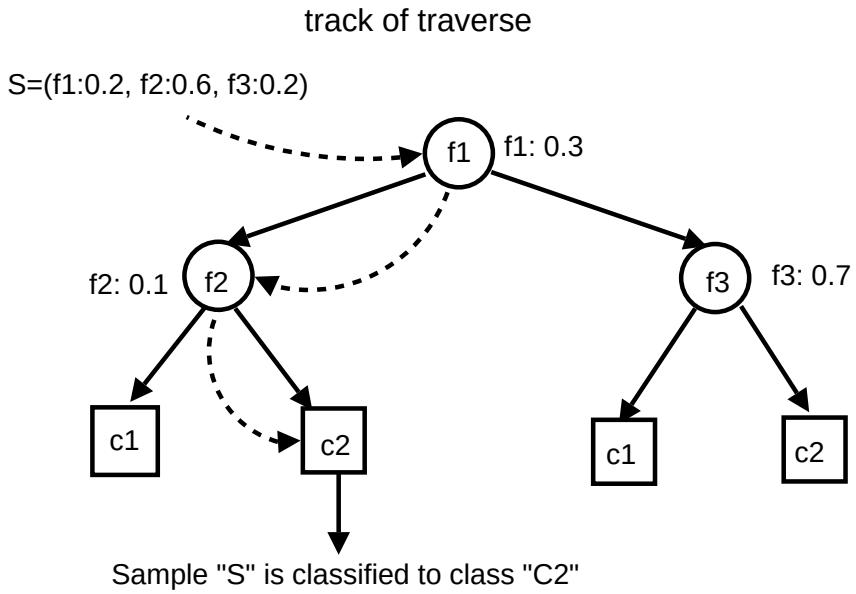


Figure 2.5: A decision tree example showing classification using three function words.

features are used in this example for illustration. In text classification, an instance is a piece of text or document, and features are tokens extracted from that document, such as words or phrases. After calculating the information gain for the three features individually, feature f_1 is chosen as the root of the tree, and feature f_2 and f_3 are the branches. Six constraints are then derived based on the three information gain values, as shown in Figure 2.5, which are used to direct a text traversing along the tree. The document S, represented as $(f_1 : 0.2, f_2 : 0.6, f_3 : 0.2)$, is to be classified to either c_1 or c_2. The classification process is effectively the process of traversing S from the root to the leaf, following the constraints computed from the information gain. By this approach, the document S is classified as an instance of class c_2.

In some cases information gain is a good measure for computing the relevance of a feature, but it is not always ideal. Information gain is often used to decide which of the features are the best for splitting or have the highest values. Then, these features can be placed close to the root of the tree. A notable problem occurs when information gain calculation is applied to features that have a large number of distinct values. Suppose we intend to build a decision tree for student records data. One of the input features might be the student number.

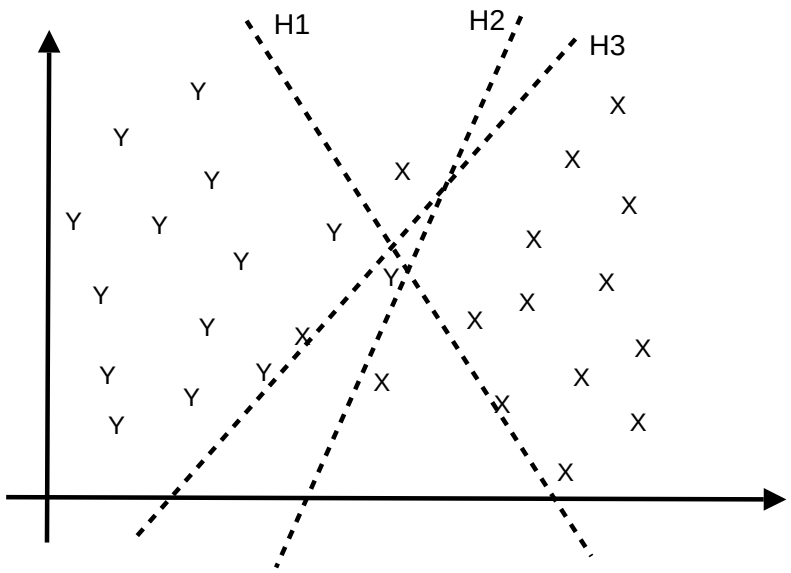


Figure 2.6: An example of a SVM for classification.

Intuitively the information gain of this feature is high due to the fact that each student has a unique student number, so this feature is likely to be placed as the root. However we do not want to include this information in the decision tree because grouping students by their student number is unlikely to generalise to new students.

Support Vector Machines

Support vector machines were introduced in 1992 and have been successfully applied to handwriting recognition and automated text categorization (TC) [Diederich et al., 2003; Joachims, 1998; Kwok, 1998; Schölkopf and Smola, 2002; Tong and Koller, 2002]. The basic principle of SVMs is to find a hyperplane separator in the feature space, usually a high dimensional feature space, that can best separate samples of different classes. The hyperplane is also referred to as the decision boundary.

An example of a SVM in a two-dimensional space is shown in Figure 2.6. As seen, there are two classes of data in the space, and the task is to classify these data into the correct classes. The dotted lines in the figure represent hyperplanes that are able to separate the data clearly. Any one of these hyperplanes would probably be acceptable, however, choosing

the best one is the ultimate aim of SVMs. In the following, we briefly review the fundamental principles of SVMs for classification [Vapnik, 1998; Joachims, 1998].

Suppose that we have built a machine to learn mappings from observations of input-output pairs $x_i \rightarrow y_i$, where the machine is in fact a function of x, given an adjustable parameter α. We use the notation $f(x, \alpha)$ to express this function. If x_i and y_i are continuous values then the true error made by this function of mapping $x_i \rightarrow y_i$ is:

$$E(\alpha) = \int \frac{1}{2} |y - f(x, \alpha)| \, dP(x, y) \tag{2.27}$$

where, $P(x, y)$ is some assumed probability distributions from which the data is drawn. However, the actual probability distribution of $P(x, y)$ is normally unknown, therefore an estimated error rate is measured on the training data only—that is, a finite number of observations. The empirical error is:

$$E'(\alpha) = \frac{1}{2l} \sum_{1}^{n} |y_i - f(x_i, \alpha)| \tag{2.28}$$

where l is the number of training instances, and there is no probability distribution $p(x, y)$ required. Therefore the E' is fixed when α and the training set are both fixed.

SVMs are instances of learning machines that can be grouped into linear SVMs and non-linear SVMs. The linear SVMs are the simplest amongst all types of SVMs. Linear SVMs are trained on separable data and non-linear SVMs are trained on non-separable data. SVMs were initially proposed for binary classification by finding a hyperplane in a high dimensional space that can best separate the data of two different classes.

All learning machines are trained by $\langle \vec{x_i}, y_i \rangle$ pairs, where $\vec{x_i} \in R^n$, meaning that each $\vec{x_i}$ is a vector of n dimensions. The values $y \in \{1, -1\}$ indicate the classes, for which 1 represents positive class and -1 the negative class. To simplify the concept, we start with a discussion of SVMs in a two-dimensional space.[6]

Suppose we have some solid line in a two-dimensional space as shown in Figure 2.7, which can separate data of two classes. The data points x on the line satisfy the equation $w \cdot x + b = 0$, where w is normal to the line. Now if we expand these concepts into a high

[6]The discussion we provided is far from thorough, which is only the principle; there are very good materials that provide more comprehensive discussions [Vapnik, 1995; 1998].

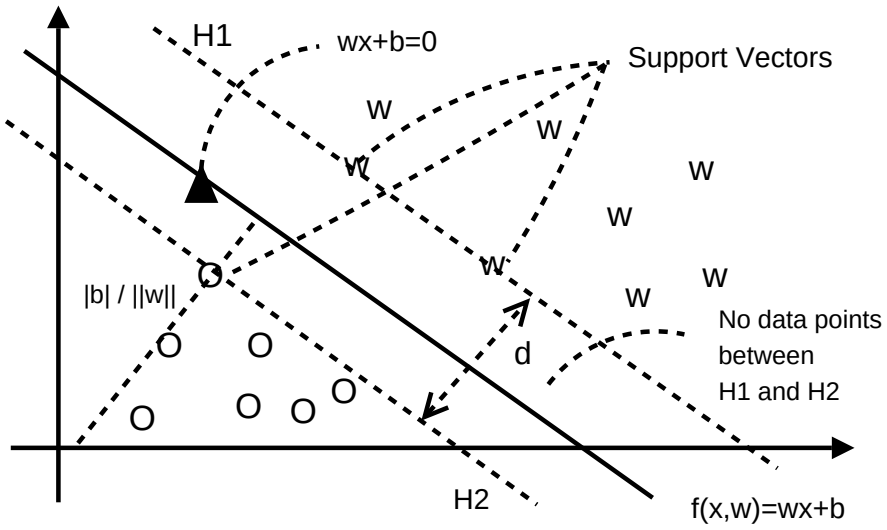


Figure 2.7: An example of linear separating hyperplane in SVMs.

dimensional space, such a hyperplane can be represented as:

$$\vec{w} \cdot \vec{x} + b = 0 \tag{2.29}$$

where w is normal to the hyperplane that can separate data points, $|b|/||w||$ is the perpendicular distance between the origin to the hyperplane, and $||w||$ is the Euclidean norm of $\vec{w}$. Given such a hyperplane, the closest positive data points and negative data points to this hyperplane are referred to as "support vectors"—that is, the data points lying on the hyperplanes that satisfy $\vec{w} \cdot \vec{x} + b = 1$ and $\vec{w} \cdot \vec{x} + b = -1$, as shown in Figure 2.7. The perpendicular distance between a positive support vector and a negative support vector is d; SVMs aim to find a hyperplane that can separate the data points, while maximising the value of d.

The output values of y_i are 1 if instances are from the positive class, or -1 if instances are negative.

$$\vec{w} \cdot \vec{x_i} + b \geqslant 1 \qquad \text{for } y_i = 1 \tag{2.30}$$

$$\vec{w} \cdot \vec{x_i} + b \leqslant -1 \qquad \text{for } y_i = -1 \tag{2.31}$$

Therefore the combined constraint based on the above two equations can be derived as:

$$y_i \left(\vec{w} \cdot \vec{x_i} + b \right) - 1 \geqslant 0 \qquad \forall i \tag{2.32}$$

Recall that all points lying on the H_1 hyperplane in Figure 2.7 satisfy $\vec{w} \cdot \vec{x_i} + \vec{b} = 1$. The perpendicular distance from H_1 to the origin is then given by $|1-b|/||w||$. Similarly, the perpendicular distance from the H_2 hyperplane to the origin is $|-1-b|/||w||$. The distance between H_1 and H_2 is also known as margin d, where $d = 2/||w||$. In this case, in order to maximize the distance d, $||w||$ is to be minimized subject to the constraint of Equation 2.32. If the support vectors are removed from the data, the solution of the problem is inevitably changed. This is a basic property of support vectors. The separating hyperplane can be found by solving the optimization problem:

$$\begin{aligned} &\textit{Minimise } \frac{1}{2}||w||^2 \qquad \text{subject to} \\ &1 - y_i \left(\vec{w} \cdot \vec{x_i} + b\right) \leqslant 0 \end{aligned}$$

Recapitulate the constrained optimisation before we solve the above problem. To simplify the formula, we use x_i to represent $\vec{x_i}$ and w to represent $\vec{w}$. Suppose we want to minimise $f(x)$ subject to constraint $g(x) \leqslant 0$, a necessary condition for x_0 to be a solution is:

$$\frac{\partial}{\partial x}\left(f(x) + \sum_i \alpha_i g_i(x)\right)|_{x=x_0} = 0 \quad \text{where} \tag{2.33}$$

$$g_i(x) \leqslant 0 \qquad \text{for } i = 1, \cdots, l \tag{2.34}$$

The function of $f(x) + \sum_i \alpha_i g_i(x)$ is also known as Lagrangian, whose gradient is to be set to 0. By definition, the gradient is a column vector whose components are the partial derivatives of the function itself. Replacing the inequality constraints and the problem function, this gives Lagrangian as:

$$\begin{aligned} L &= \frac{1}{2}||w||^2 + \sum_1^n \alpha_i \left(1 - y_i \left(wx_i + b\right)\right) \\ &= \frac{1}{2}||w||^2 - \sum_1^n \alpha_i \, y_i \left(x_i \cdot w + b\right) + \sum_1^n \alpha_i \end{aligned} \tag{2.35}$$

L must be minimised with respect to w and b, and maximised with all $\alpha \geqslant 0$.

$$\begin{aligned} \frac{\partial L}{\partial w} &= \frac{1}{2}w + \sum_1^n \alpha_i \left(-y_i\right) x_i = 0 \quad \Rightarrow \\ w &= \sum_i \alpha_i y_i x_i \end{aligned} \tag{2.36}$$

and

$$\begin{aligned} \frac{\partial L}{\partial b} &= 0 \quad \Rightarrow \\ \sum_i \alpha_i y_i &= 0 \end{aligned} \tag{2.37}$$

If we substitute Equations 2.36 and 2.37 back to Equation 2.35:

$$L' = \sum_1^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{2.38}$$

Therefore the problem is now cast in terms of α_i only. This is known as the dual problem, meaning that if we know w, then we are able to know all α_i, and vice versa; while the original problem is known as the primal problem. Both L and L' are derived from the same objective function but with different constraints.

Now the dual problem of $\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$ is to be maximised, subject to the constraint of Equation 2.37. When an optimal separator is found, then the weights of α_i associated with many data points are zero, except those which are closest to the separator. These points with non-zero parameters hold up the hyperplanes for separating classes. In other words, all the support vectors lie on either hyperplane H_1 or H_2 and the corresponding $\alpha_i > 0$. If a data point locates in one side of either H_1 or H_2 then it is assigned to the corresponding class.

The ideal case is that a linear boundary can be found. However linear boundaries are hard to find in practice, because the data is not linearly separable in most cases. An alternative way is to define a kernel function to map data from an input space to a feature space; after a proper transformation the classification can become easier. (An example is shown in Figure 2.8). There are several kenerl functions have been proposed and applied to a range of applications, both linear and non-linear [Joachims, 1998; Vapnik, 1998; Diederich et al., 2003].

One of the biggest limitations in the use of SVMs is the difficulty of choosing an appropriate kernel function. Identifying the best kernel function for a given data set remains a challenge. Another limitation is the efficiency; the best algorithm proposed for optimisation in SVMs has computational complexity of $O\left(kn^2\right)$, where k refers to the dimensionality of the feature space and n is the number of training samples. This indicates that for large n, the

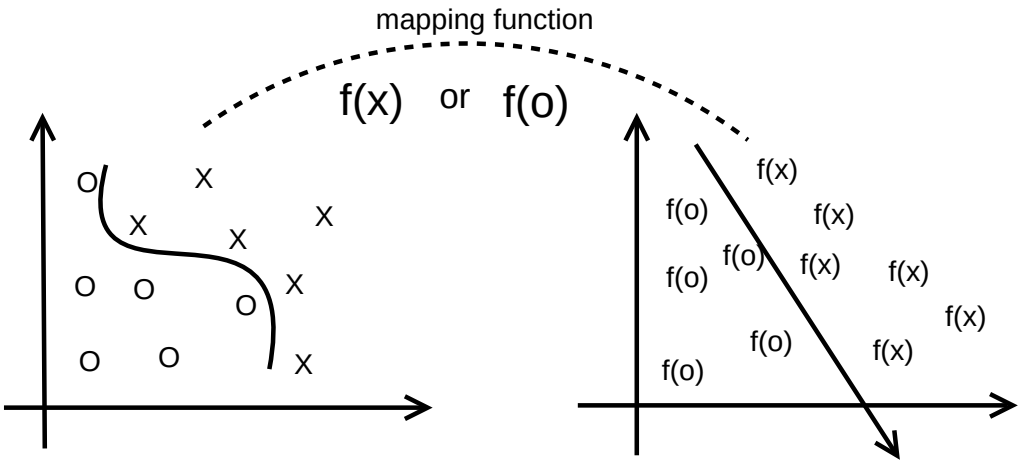


Figure 2.8: An example of mapping non-linear separable data to liner separable data in another feature space.

computational cost can increase dramatically. Additionally, SVMs are not straightforward for multi-class classification; a commonly used transformation is to convert a multi-class problem into a set of binary classification problems.

SVMs were first applied to TC by Joachims [1998]. In recent years, SVMs have enjoyed considerable attention in TC and have been widely investigated [Drucker et al., 1999; Dumais et al., 1998; Klinkenberg and Joachims, 2000; Masuyama and Nakagawa, 2004; Moschitti and Basili, 2004; Sassano, 2003]. As suggested by Joachims [1998], SVMs can be robust to overfitting, and can scale up to a fairly high dimensionality. In TC, SVMs usually require large numbers of samples for training in order to achieve satisfactory effectiveness [Joachims, 1998]. Therefore for attribution problems with limited data, SVMs are not always superior to other learning methods. Sassano [2003] has proposed a method of creating virtual support vectors to improve the effectiveness of SVMs when the training data is limited. Virtual samples were generated from the labelled training examples, by adding or deleting words from some of the original sample.

Machine learning approaches have been the state-of-art approaches in the field of automated TC. The selected methods have been shown to be competitive for text classification, nonetheless, SVMs are the most effective learning approaches for many text classification tasks, in particular for collections of large size. TC techniques are valuable for other research

problems as well—authorship attribution is one of these tasks.

2.3 Authorship Attribution

Authorship attribution (AA), as the name implies, is concerned with determining the authors of disputed texts or texts with missing authorship. It is a categorization task on textual data, and therefore, research such as automated text categorization provides much of value for AA investigations.

Generally speaking, automated AA employs computational linguistic techniques rather than relying on external evidence obtained from the original manuscripts, such as handwriting and signatures. Automated AA is of great importance and may be potentially used in a range of applications. First, AA can be used in the purely scholarly sense: for instance to examine whether the works of Shakespeare were actually written by someone else,[7] or to find out who wrote the 12 anonymous *Federalist Papers* [Fung, 2003; Khmelev and Tweedie, 2002]. Second, AA can be used in forensic analysis for criminal investigations [Foster, 2000]. Also, AA can be used for plagiarism detection, and related areas of investigation.

In the field of AA, there are three kinds of evidence that can be used to establish authorship: external, interpretive, and linguistic [Carin, 1998]. External evidence includes an author's handwriting or signatures in the original manuscripts; however these kinds of features are rarely used by automated AA systems. Interpretive evidence is the study of what the author meant when a document was written. This requires experts, and such obtained evidence is very subjective, and may vary greatly from one expert to others. Linguistic evidence focuses on the patterns of word usages that can be observed in documents. Current automated AA investigations focus on this type of linguistic evidence.

Any AA system starts with a set of training documents that have identifiable authorship (not collaborative). Style-bearing features are extracted from the training documents, and a classification method is then applied to these extracted features. Given a set of target authors available in the collection $A = \{a_1, a_2 \dots a_n\}$, there are in general three types of AA problems based on the cardinality of $|A|$: binary or two-class AA with $|A| = 2$, multi-class AA with $|A| > 2$, and one-class AA with $|A| = 1$.

[7] See for example shakespeareauthorship.com

The aim of binary AA is to assign all documents to either one of two target authors. In this problem all documents in the data set are written by one of the two authors, even the documents that are to be attributed. Binary AA is the simplest classification problem [Binongo, 2003; Fung, 2003; Holmes et al., 2001]. In multi-class AA, more than two potential authors ($|A| > 2$) are to be differentiated from each other [Baayen et al., 2002; Diederich et al., 2003; Juola and Baayen, 2003]. The more potential authors that are included, the harder the attribution task it is. One-class AA is also referred to as authorship verification [Koppel and Schler, 2004]. Documents are to be identified as either written by a particular author or not. In contrast to other AA tasks, it has not been widely examined. There is no prior knowledge of how many distinct authors are included in the collection, or how many documents each author has. In this case, it is intuitively difficult to generalise the writing style for each of the authors individually. Documents written by the target author in the collection are known as positive samples, while others are negative samples. Note that collections for one-class AA are skewed in most cases—that is, the number of negative documents is usually much larger than the number of positive samples. It is easier to capture and generalise writing habits for a particular author rather than for "not-the-author".

2.3.1 Stylometry

Stylometry makes the fundamental assumption that authors have distinct writing habits, examples include, the richness of an author's core vocabulary usage, the complexity of the sentences on average, as well as the rhythm and flow [Holmes, 1998]. It is further assumed that these habits are difficult to be disguised consciously. Authorship attribution (AA) is such an application based on these assumptions. The study of AA focuses on two aspects: features and attribution methods. Given a document, features that are able to reflect styles of writing in a certain respect should be extracted. An attribution method is then applied to measure these features in order to differentiate between authors.

Authorship attribution is effectively a process of partitioning texts or documents. In this respect, techniques of AA share a similar framework to that of text categorization (TC). However, due to the different partitioning criteria being involved, the techniques being used are significantly different. In TC, the grouping of documents is based on the topics or the

content of the documents. Therefore, it is straightforward to use content words as features for document representations. The hypothesis is that the more a content word occurs in a document, the more substantial it is in terms of the semantic meanings or the topics for that document. Function words are usually removed from documents; these are words that have little semantic meaning of their own but grammatically important, such as "a", "of", and "the". All remaining distinct words in a collection are used as features; these words are in contrast considered as non-common words, such as "fish" and "medicine".

In contrast to TC where categorization of texts are based on the subject matter, in AA texts are partitioned based on the authorship or styles of writing. Additionally, a finite set of stylometric features (or style markers) are pre-defined for AA, while pre-defining such a set of features is not practical in TC. Whether topic words are plausible for AA has been controversial. Some scholars argue that even with using topic words describing the same event, the choices of words may be sufficiently different. Also, eliminating content words may cause information loss and degrade the effectiveness of AA [Burrows, 1992; Diederich et al., 2003; Holmes et al., 2001; Kaster et al., 2005; Coyotl-Morales et al., 2006]. However as we show in our experiments later in this thesis, topic words are not always plausible for AA and, are misleading, particularly with large collections. For instance, if two authors write articles about the same event, it is likely that many content words may be shared by their articles. If an AA method uses content words to differentiate the documents between these two authors, then both documents are likely to be assigned to the same authorship. This is clearly a misleading outcome caused by the use of topic words. On the other hand, some researchers argue that topic words should be retained in use for AA purposes.

The earliest studies of AA were reported by Mendenhall [1887] and Yule [1938], in which statistical methods were used limit data, not only the size of the experimental corpus but also the size of feature set. Mendenhall [1887] graphically represented the word-length as characteristic curves, and Yule [1938] used sentence length to differentiate between authors. The results suggested that these types of features are not reliable. In more recent studies, more types of features have been proposed, mainly lexical features and grammatical features.

Lexical Features

Lexical features, or style markers, refer to elements that can be extracted from the surface of texts. In AA, many types of lexical features have been proposed [Holmes, 1985; Baayen et al., 2002; Diederich et al., 2003; Holmes et al., 2001; Juola and Baayen, 2003], including:

- Token-level style markers, such as word-length, sentence-length, average sentence length;
- The frequency of word usage, such as function words; the distribution of word frequencies and punctuation frequencies;
- The richness of the vocabulary, including the distribution of vocabulary, the number of "hapax legomena" (words that are used only once in any text), and of "hapax dislegomena" (words that are used twice in a text).

These kinds of features are not difficult to extract for document representations, however some prior research has criticised such features as being not always reliable [Malyutov, 2004; Stamatatos et al., 2001; 1999]. Burrows [2002] extracted commonly used words from the collection as the features; he used the most 30 to 150 common words in the study; this feature selection was also used in his later work [Burrows, 2006]. Stamatatos et al. [1999] have experimented with vocabulary richness. The results showed that although the vocabulary richness was shown to be more informative in defining authors' writing style, it tends to be highly dependent on text length and is fairly unstable for texts shorter than approximately 1,000 words. In their more recent study, Stamatatos et al. [2001] also pointed out that merely using features at the token-level may not be sufficient for reliable AA; instead, such features are suggested to be used as a complement to other richer features.

Function words have been the most popular style markers in authorship attribution (AA); prepositions, conjunctions, and articles are all examples of function words. A straightforward reason to select function words as style markers is that they are free of topic. In other words, the usage of these words is influenced more by writing style rather than by the document content. For example, some rare function words, such as "notwithstanding", may be an indicator of the authorship as they are not commonly used in general writing. For even the commonly used function words, usage is distinguishable between authors, as we show

later. Also, prior studies of AA found that the usage of such kind of words are usually not under authors' conscious control [Holmes, 1994]. Burrows [1987] first proposed the use of function words as style markers for AA. Since then function words have been widely used by many researchers. Baayen et al. [2002] experimented with 42 common function words and eight punctuation symbols; these common words were used the most frequently in the data collection used. A set of 50 common function words were selected as style markers by Holmes et al. [2001] in order to discriminate between two authors on disputed journal articles. Binongo [2003] also used 50 common function words to examine the authorship of the 15th book of Oz (whether these are the same function words is unclear). More function words have been used by Juola and Baayen [2003], in which a list of 164 function words was collected as features. These words were the most commonly used in their text collection. More recently, Pol [2005] has carried out experiments comparing the discrimination power of different lexical features, such as function words, percentage of hapax legomena, commonly used topic words, and combination of all features.

All of the aforementioned works have suggested that function words are plausible style markers for AA, however the conclusions are not necessary reliable due to the limited data collections that were used. The effectiveness and scalability of function words need to be further evaluated. A further problem is that different sets of function words were applied with different collections. The lack of consensus about which are better function words for AA has led difficulties in comparing these AA approaches, although the research does suggest that function words are feasible style markers.

Linguistic Features

Syntactical or grammatical components can be successfully extracted by using natural language processing (NLP). These are considered as grammatical-based or syntax-based features in AA, which have been applied by several researchers [Baayen et al., 1996; Kukushkina et al., 2001; Stamatatos et al., 1999]. NLP has been widely investigated for applications such as machine translation and word alignment. In the area of AA, NLP is often used to annotate collections with grammatical components by using features such as part-of-speech tags, noun phrases, and sentence structures [Manning and Schütze, 1999].

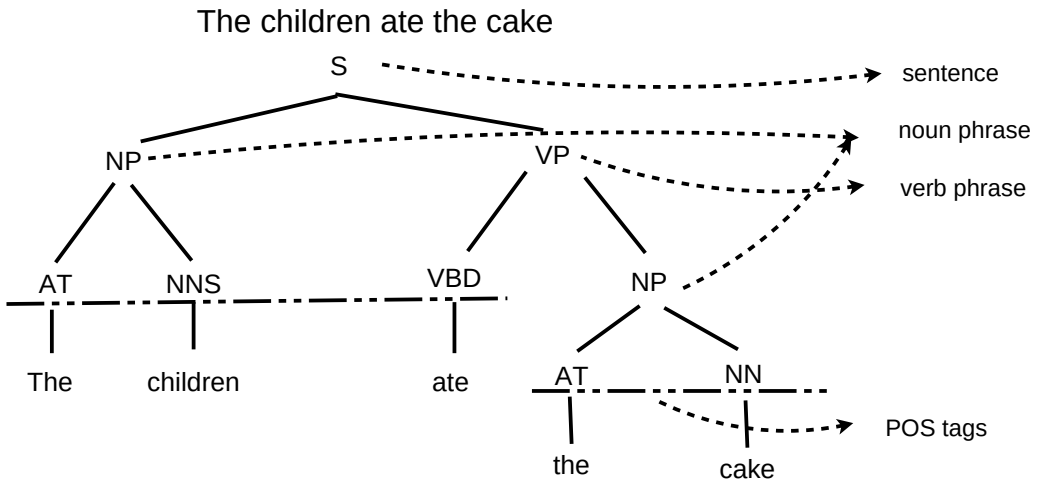


Figure 2.9: An example of grammatical structure of a sentence: The children ate the cake. The structure is represented as a tree, "Brown" tags are used.

Tagging is the process of annotating each word with its part-of-speech (POS) tag. POS tags are lexical categories. Broadly speaking, linguists recognize four major categories of words in English: nouns, verbs, adjectives, and adverbs. Each of these types can be further classified according to morphology. Most part-of-speech tag sets share the same basic categories. However, different tag sets differ in terms of how finely words are divided into categories, and in how categories are defined. The "Brown tag set" has been one of the most influential tag sets, which was initially used to annotate the Brown corpus.[8]

Chunking is another popular annotation in AA. The purpose of chunking is to recognise higher level units of structure in order to compress sentence descriptions. The basic description consists of prepositional phrases, verb phrases, noun phrases, and so forth. An example of analysing syntax-based features is shown in Figure 2.9. In terms of popularity in recent AA investigations, the syntax-based features are less common than simple lexical features for several reasons. First, the extraction of such kinds of features relies on NLP techniques, and therefore the effectiveness of AA using these features is inevitably subject to the goodness of the NLP methods used. Second, the syntax-based linguistic features are usually

[8]The "Brown tag set" is applied through this thesis. Also, there are other tag sets: "c5 tag set" is used for tagging British National Corpus; "Penn Treebank tag set" is a simplified version of the "Brown tag set" that has been widely used in NLP [Manning and Schütze, 1999].

Table 2.3: An example of rewrite rules that are based on analysis of the sentence shown in Figure 2.9, "The children ate the cake".

left POS		rewritten rules
S	$\rightarrow$	$NP\ VP$
NP	$\rightarrow$	$AT\ NNS \mid AT\ NN$
VP	$\rightarrow$	$VBD\ NP$

difficult and expensive to compute. There are a variety of NLP tools with diverse functionalities, for instance, TOSCA [Baayen et al., 1996; Oostdijk, 1991], CCPP [Keulen, 1986], SCBD [Stamatatos et al., 1999; 2001], NLTK [Bird, 2006], and Stanford Lexparser [Kaster et al., 2005]. Almost all of these NLP tools can perform tagging and chunking, but with great differences in methodologies, output, and effectiveness. However, several researchers have suggested that NLP can be a plausible source of alternative features for AA.

Baayen et al. [1996] have used TOSCA [Oostdijk, 1991] to annotate texts with syntactic markers; that is, a set of rewrite rules were extracted as features from the collection. A rewrite rule has the form of "category $\rightarrow$ category*", the symbol on the left hand side of which can be rewritten as symbol sequence on the right hand side. Here, "category" usually refers to POS tags. Taking the sentence in Figure 2.9 for example, some simple rewrite rules can be extracted, as listed in Table 2.3. These rules solely depend on the categories of words, not on any surrounding context, and thus can be regarded as a plausible choice of style marker. Their study indicates that this kind of syntactic annotation is at least as effective as lexical-based features.

Stamatatos et al. [1999; 2001] have used SCBD, another NLP tool, to detect the boundaries between sentences and chunks of unrestricted modern Greek texts, a collection of newswire articles. Features were extracted from the output of SCBD. In their work, a set of 22 style markers were defined on three stylometric levels: three on the "token-level", ten on the "phrase-level", and nine on the "analysis-level". To extract token-level features, the input text was considered as a sequence of tokens. For extracting phrase-level features, the input text was considered as a sequence of phrases or chunks ("NP", "VP" and so on as mentioned previously). The analysis-level features were more complicated, containing in-

formation that could not be reliably represented at the first two levels [Stamatatos et al., 2001]. Surprisingly, the three token-level features have produced the highest accuracy of 61%, while only 50% accuracy has been achieved by using phrase-level features, and 55% accuracy by using analysis-level features. Combining features on the higher level with features on the token-level was able to increase the accuracy up to 81%. In addition, a further 6% improvement was achieved by considering a number of 50 commonly used topic words.

More recently, Kaster et al. [2005] experimented with linguistic features, including POS tags and syntax tress; the results were compared to the baseline set by using bags-of-words. The results suggested that simple bags-of-words performed better than features from the richer syntax trees, while a combination of both produced significant improvement in AA effectiveness. The combined features were reported to be able to achieve an accuracy higher than 85%. However the results may not be convincing and the conclusion may not be comprehensive, due to the corpus used in this investigation. The books and authors selected were obtained from Project Gutenberg, where books are highly duplicated, as we show later in Chapter 7. The numbers of books used in their work are much larger than the distinct books for certain authors. Intuitively, duplicated books can be identified correctly with little difficulty by any classification approaches; this can easily cause an over-estimate of the accuracy.

In AA, many different types of features have been tried, however the reported results are not directly comparable. It is difficult to draw a comprehensive conclusion on which are the "better" features for AA purposes. One of the key issues is the diversity of test collections, where it is almost always the case that each researcher has their own collection that supports the success of experimented features and methods. The robustness and scalability of the existing methodologies remains unclear. In the following section, we review the data collections that have been used in AA.

2.3.2 Collections in Authorship Attribution

Recall that, one of the major challenges in AA is lack of benchmarks, and thus, lack of baselines. Without such benchmarks it is difficult to compare the results of previous research in the literature. Therefore, there is no basis for claiming which AA techniques are better

than others.

A wide range of data collections have been used to evaluate AA approaches, texts of which are collected from different information domains:

- Shakespeare plays: The Shakespeare authorship debates is a historical issue that has lasted for centuries. Serval pioneer studies in the field of AA mainly used these texts [Mitchell, 1996; Williams, 1975].

- Federalist papers: another widely used collection in early AA research [Baayen et al., 1996; Holmes and Forsyth, 1995; Malyutov, 2004; Mosteller and Wallace, 1964; Khmelev and Tweedie, 2002; Kjell and Frieder, 1992; Kjell, 1994a;b; Tweedie et al., 1996]. The Federalist papers were written from 1787 to 1788. There were 85 texts in total, of which 52 were believed to be written solely by Hamilton and 14 solely by Madison. There are 12 whose authorship is disputed. A further 13 texts were jointly written by both authors, and the remaining four texts were written by Jay, which were not considered in most AA studies.

- English literature: These collections contain novels in English. For example, Binongo [2003] has used a collection of 15 Oz books, of which 14 books were used for training to determine the authorship of the 15th Oz book. Baayen et al. [1996] used two works of fictions for binary AA. The same texts were also used by Khmelev and Tweedie [2002]. Many researchers also created data from Project Gutenberg,[9] choosing different books [Kaster et al., 2005; Khmelev and Tweedie, 2002; Luyckx et al., 2006]. Hoover [2001] has used 46 British and American novels by 31 authors. A collection of 21 nineteenth century English books by 10 different authors were used by Koppel and Schler [2004], spanning a variety of genres.

- Non-English materials: 90 Italian books by 11 authors have been used by Benedetto et al. [2002]. Luyckx and Daelemans [2005] have used documents of two authors, which were taken from the online archive of the Belgian daily newspaper *De Standaard*. Each author has 100 articles for training and 34 articles for testing. Diederich et al. [2003]

[9] http://www.gutenberg.org

collected texts from the Berliner Zeitung, a daily newspaper in Berlin, from December 1998 to February 1999. Kukushkina et al. [2001] used 385 Russian texts by 82 writers.

- English newswire articles and journals: articles from 50 journalists were used by Sanderson and Guenter [2006].

- Students' essays: A collection of 72 Dutch articles were collected by Baayen et al. [2002]. They were written by 8 university students of Dutch literature, each of whom had written three essays on each of the three genres: fiction, argument, and description. The same collection has been used by Juola et al. [2006].

- Electronic messages: A total number of 156 emails from three authors have been compiled by Vel et al. [2001]. Argamon et al. [2003] have used a collection of 500 newsgroup posting threads, consisting of the 10 most frequent authors as well as the 10 least frequent authors.[10] Koppel and Schler [2003] have used a corpus of 480 emails that were written by 11 authors during a period of one year. These emails were written on three topics: movies, food, and travel.

- Poems: A collection of 353 poems written by 5 authors was gathered by Coyotl-Morales et al. [2006]. Burrows [2002] used a collection of poems from 25 poets from English Restoration era and Burrows [2006] used eight poems from the English Restoration era. Three of them have unquestioned authorship, and the other five are disputed.

None of these collections has been made available as a standard benchmark for AA, and there has been little comparison of methods. The reported successful methods are not guaranteed to be successful for handling other data collections, even for document collections containing the same type of texts [Goodman, 2002]. For example, Goodman [2002] has failed to reproduce the results by Benedetto et al. [2002], using the same method on different data sets.

Further problems can be observed with the collections used; the collections are generally small in earlier AA research. Holmes et al. [2001] used only 17 journal articles written by two journalists. Only one out of the 17 was classified, and there is no evaluation with the other

[10] Messages are collected from http://groups.google.com

16 books. Similarly, Binongo [2003] used 14 *Oz* books to predict the author for the 15th *Oz* book. Baayen et al. [1996] only used 2 books. The collection used by Baayen et al. [2002] consists of 72 unedited tests of 8 Dutch students. The average length of these documents is around 1,000 words, which is relatively short. The collection of Federalist paper is also small, 65 texts in total. In this case, results with only one or two different decisions could lead to big numeric differences in accuracy. On the other hand, some researchers have used larger collections in recent years. For instance, the number of samples used by Argamon et al. [2003] was 500 by 20 authors; Koppel and Schler [2003] used 480 emails from 11 authors; Kukushkina et al. [2001] used a collection of 385 texts, by 82 writers; and, Diederich et al. [2003] built a corpus of approximately 700 neswire articles from seven authors (around 100 documents per author). Although these collections contain more documents and authors, they are not particularly big. The scalability of AA approaches is hard to examine by using data collections of small sizes, and the effectiveness is not reliable either. Therefore, developing suitable collections is of great importance in the field of AA.

Also note that the collections are derived from different knowledge domains or in languages other than English, and therefore, techniques that work for one data set may not be effective for others. Many existing investigations in the AA literature were based on a single data set, or specific authors.

2.4 Existing Approaches for Authorship Attribution

Once collections and features are selected, some classification methods are then applied for attribution. The evaluation metrics used in AA are similar to those used in TC; accuracy and error rate are commonly used, as defined in Section 2.2.3. We review current AA techniques in following sections, from statistical and computational methods, to machine learning approaches.

2.4.1 Simple Statistics Measures

Early research in authorship attribution (AA) usually involved using simple statistics of writing patterns in given documents. For instance, the numbers of words used once, twice, and so forth were counted to analyze the writing style of Shakespeare by Efron and Thisted

[1976]. The results suggested that, if a new work of Shakespeare were discovered, it would contain a certain number of words that had never been used in any of the known works. Smith [1983] used lexical features, such as average word length, average sentence length, and collocations, with the Chi-square (χ^2) statistical significance test used to differentiate between Shakespeare and Marlowe. Chi-square is formulated as:

$$\chi^2 = \sum_i \frac{(O_i - E)^2}{E} \tag{2.39}$$

where E is the expected values of features, and O_i indicates the observed values. He suggested that neither word length nor sentence length is reliable; both are likely to give incorrect predictions. Recently, the Chi-square (χ^2) measure is often used to determine relevant features in applications such as text categorization [Yang and Pedersen, 1997], and authorship attribution [Kaster et al., 2005; Stamatatos et al., 2001].

The cusum (cumulative sum) technique looks at the frequencies of a range of possible habits in use of language; a detailed description is given by Farringdon [1996]. The assumption made in use of cusum technique is that the patterns of using short words as well as words beginning with a vowel are characteristic and can be used to discriminate between authors. It plots the cumulative sum of differences between the observed short word counts in the given documents. It is claimed that only five sentences in a sample text of unknown authorship are required to be tested against the texts of known authorship. This statement is particularly significant for forensic investigation. This kind of technique is supported by many researchers, including Lohrey [1991], Storey [1993], and Canter and Chester [1997]. However, it has been criticised by De-Haan [1998] and Hardcastle [1997], who have shown unreliable results using this cusum approach.

More recent research attempted to identify the "best" stylometric patterns, as well as to apply more sensitive classification methods rather than simple statistical measures. Principal component analysis (PCA) [Baayen et al., 1996; 2002; Holmes et al., 2001; Burrows, 2002], Markov chains [Khmelev and Tweedie, 2002; Khmelev and Teahan, 2003b], and compression-based techniques [Kešelj et al., 2003; Peng et al., 2003a] are typical of computational approaches that have been proposed for authorship attribution (AA).

2.4.2 Principal Component Analysis

There are many techniques that fit under the category of PCA, which is a statistical approach based on the theory of matrix algebra and is also considered as a clustering method sometimes. PCA has been reported with successful results for many earlier studies in AA.

The basic idea underlying PCA for AA is mining the most significant patterns from all possible patterns [Rencher, 2002]. Principals are mined featuresthat can be used to predict the author of a new document. These features are also known as "principal components". Given a collection of documents $D = \{d_1, \ldots, d_n\}$ and a set of stylometric features $F = \{f_1, \ldots, f_m\}$, PCA treats the entire collection as an $n \times m$ matrix M_D that consists of n document vectors of m dimensions:

$$\begin{pmatrix} M(d_1, f_1) & M(d_1, f_2) & \ldots & M(d_1, f_m) \\ M(d_2, f_1) & M(d_1, f_2) & \ldots & M(d_1, f_m) \\ \ldots & \ldots & \ldots & \ldots \\ M(d_n, f_1) & M(d_1, f_2) & \ldots & M(d_1, f_m) \end{pmatrix}$$

where $M(d_i, f_j)$ indicates the value of feature f_j of document d_i. In order to capture the internal relation amongst different features, the covariance matrix $Cov(M_D)$ is then generated for the above matrix M_D:

$$\begin{pmatrix} Cov(f_1, f_1) & Cov(f_1, f_2) & \ldots & Cov(f_1, f_m) \\ Cov(f_2, f_1) & Cov(f_2, f_2) & \ldots & Cov(f_2, f_m) \\ \ldots & & & \\ Cov(f_m, f_1) & Cov(f_m, f_2) & \ldots & Cov(f_m, f_m) \end{pmatrix}$$

It is a matrix of $m \times m$; each value in the covariance matrix can be calculated by:

$$Cov(f_i, f_j) = \frac{\sum_n (f_i - \overline{f_i})(f_j - \overline{f_j})}{(n-1)}$$

where $\overline{f_i}$ and $\overline{f_j}$ are the mean values of feature f_i and f_j, that can be calculated from the ith and jth columns of the matrix M_D. Unlike the original matrix that consists of purely raw observations of instances, the covariance matrix presents the relationships between features based on the observations.

Calculating eigenvectors and eigenvalues of $Cov\,(M_D)$ is the key in use of PCA. Suppose λ is an eigenvalue of $Cov\,(M_D)$, the corresponding eigenvector to λ satisfies the constraint:

$$Cov\,(M_D) \times \vec{v} = \lambda \times \vec{v} \tag{2.40}$$

There are m eigenvectors $\{\vec{v_1}, \ldots, \vec{v_m}\}$ corresponding to m eigenvalues $\{\lambda_1, \ldots, \lambda_m\}$ for an $m \times m$ matrix. Eigenvectors of a matrix are perpendicular to each other; they are considered as the patterns of the data. The m eigenvectors are known as the components of the collection D. In theory the eigenvectors with the highest eigenvalues are considered as the stronger patterns of the data, also known as the principal components.

A new feature set F' is then generated based on the selected principal components. It is the new representation of the original data, collection D. If all m eigenvalues and eigenvectors are kept, the matrix of the new feature set is then:

$$F'\,(D) = \left(\vec{v_1} \ \ldots \ \vec{v_m} \right) \tag{2.41}$$

In almost all previous AA studies based on PCA, only the first two principal components are kept for classification; less significant components are usually neglected. These two principal components can be plotted graphically in a two-dimensional space, where the first principal component is plotted against the second [Baayen et al., 1996; Binongo, 2003; Burrows, 1987; Holmes et al., 2001]. The pattern can then be observed from the data clusters. In the context of AA, documents in the same cluster share the same authorship.

Burrows [1992] has analyzed the frequencies of 50 commonly occurring words in the texts being examined. He used PCA to plot as a graph of the first component against the second. The results showed that the data points can be clearly separated. The results also suggested that PCA is a good technique for visualization of differences between authorship, and can effectively lower the dimensionality of the original data to two or three dimensions only.

Baayen et al. [1996] experimented with a syntactically annotated corpus using PCA. Two crime novels were selected as the test collection; both novels were broken into 2,000-word chunks. The 50 most frequent words and the 50 most frequent rewrite rules were extracted as features. The rewrite rules were formed in the similar way to that described in Section 2.3.1. Frequencies of features were calculated for individual documents in the collection. The results

showed that PCA can effectively distinguish between two authors. Zero-misclassification was reported as the best result by applying rewrite rules.

Binongo [2003] has applied PCA to a list of 50 commonly used function words as features to examine the authorship of the 15th book of Oz. L. Frank Baum was famous for writing the books of *Wonderful Wizard of Oz*, and produced a series consisting of 14 books. However, the 15th book in the series is now believed by many people to be R. Plumly Thompson's work. In the experiment by Binongo [2003], the first 14 books were compared with the 15th book; the statistical analysis by PCA showed higher compatibility with Thompson's than with Baum's.

Holmes et al. [2001] have applied both traditional and non-traditional methods of AA to identify a collection of 17 journal articles of uncertain authorship. These articles were published in the *New York Tribune* between 1889 and 1892 with the claimed authorship of Stephen Crane.[11] The set of 50 common words was the same as used by Burrows [1992]. The results showed that the non-traditional AA approach can provide statistical evidence to complement traditional AA. It was again effectively a binary classification problem. Note that it is often the case that PCA has been applied to distinguish between two authors, and moreover, the size of data collections is usually small.

Although success has been reported, PCA is not ideal for AA. Hoover [2001] has experimented with PCA and suggested that it was neither scalable nor effective for reasonably large data collections, or for multi-class AA. He has used 3,000-word chunks from 50 novels of mixed genres that were written by 27 authors. A set of 50 commonly used function words were used as features. However, only 25% accuracy was achieved by PCA. Surprisingly, the effectiveness has been degraded by a further 3% when increasing the number of function words to 100. In contrast, reducing the number of authors from 27 to 10 boosted the effectiveness to 60%. Increasing the number of words included in the segments also improved the effectiveness. The results have suggested that PCA may often not be accurate enough to determine the authorship of a given text. It is relatively effective when the number of authors is very small, in most cases, only two. Increasing the number of features does not guarantee

[11] Stephen Crane was a nineteenth century American writer. He also worked as a journalist for the *New York Tribune*.

a higher accuracy. On the other hand, the number of discarded components is far more than those being used. In this case, information loss can be high, in particular, those discarded components may be also informative for mining authors' writing styles. Last, interpretation of the output of PCA is not straightforward, usually requiring extra measures.

2.4.3 N-grams and Markov Chains

N-grams are widely used in language models, which were originally developed for speech recognition [Jurafsky and Martin, 2000] and have been widely used in a variety of applications in recent years, including text categorization [Peng et al., 2003a], information retrieval [Gao et al., 2004; Lafferty and Zhai, 2001], and authorship attribution [Khmelev and Tweedie, 2002; Kukushkina et al., 2001; Peng et al., 2003a].

N-gram based approaches can operate at either the word level or character level. In the use of such techniques, a document or a piece of text is regarded as a sequence of n words (or characters), $W = \{w_1 w_2 \dots w_n\}$, where n is the number of words (or characters) in this document. The probability of occurrence of the sequence W is derived as:

$$p(W = w_1 w_2 \dots w_n) = p(w_n | w_1 \dots w_{n-1}) \quad \text{applying chain rule:} \tag{2.42}$$

$$= \Pi_{i=1}^{i=n} p(w_i | w_1 \dots w_{i-1}) \tag{2.43}$$

An N-gram is a unit formed by a total of N consecutive words or characters, thus, a document is considered as a set of N-grams. The above calculation can then be applied to compute the probability of each N-gram occurring in the document, conditional on its preceding $N-1$ words or characters. The simplest N-gram is uni-gram, where $N = 1$. In such a uni-gram model, there is an independence assumption made. The assumption states that each word occurring in the document is independent of its preceding words:

$$p(w_i | w_1 \dots w_{i-1}) = p(w_i) \tag{2.44}$$

$$p(W = w_1 w_2 \dots w_n) = \Pi_{i=1}^{i=n} p(w_i) \tag{2.45}$$

A straightforward estimation of N-gram probabilities is the maximum likelihood estimate. The probability of a word in a document is given by the frequency of that word in the

document normalized by the total number of words in that document:

$$p(w_i) = \frac{f_{w_i,d}}{|d|} \tag{2.46}$$

where $f_{w_i,d}$ indicates the frequency of term w_i in the document d. Substituting Equation 2.46 into Equation 2.45, the probability of a word sequence W, or a document, generated by a uni-gram model can be calculated as:

$$p(w_1 \dots w_n) = \Pi_{i=1}^{i=n} \frac{f_{w_i,d}}{|d|} \tag{2.47}$$

More complicated models can be constructed with $N > 1$. For example, in a bi-gram model, also known as first-order Markov chain, N equals 2; the probability of a term occurring in a document is conditional on one single preceding term. The model is formulated as:

$$p(w_i|w_1 \dots w_{i-1}) = p(w_i|w_{i-1}) \tag{2.48}$$

$$p(w_i) = \frac{f_{w_{i-1}w_i,d}}{f_{w_{i-1},d}} \tag{2.49}$$

from which the probabilities of word sequences W generated by a bi-gram model can be derived:

$$p(W = w_1 \dots w_n) = \Pi_{i=1}^{i=n} \frac{f_{w_{i-1}w_i,d}}{f_{w_{i-1},d}} \tag{2.50}$$

Models can be built on the character level in the same manner, where each feature f indicates a single character rather than a single word.

In AA, first-order Markov chains have been employed. Given a set of potential authors to be differentiated, $A = \{a_1, a_2 \dots a_n\}$, each of them has written several texts in the collection. An $m \times m$ transition matrix can be constructed for training sets associated with each of the authors. Here, m refers to the number of features that are defined. Each position (i, j) in the matrix records the number of transitions from character i to j in a document (character i occurs before j).

Now, we use $Q_{ij}^{a,d}$ to represent the weight (usually the probabilities that can be measured by N-gram models as presented previously) of $i \rightarrow j$ transitions in the document d that is written by author a. In order to predict the actual author for a unknown document d', a model should be built on each training set provided by the potential authors:

$$Q_{ij}^{a} = \sum_{d \in D} Q_{ij}^{a,d} \tag{2.51}$$

where the computed value can be interpreted as the trained pattern of author a, using the character i before j. For each document-author pair, the probability of a being the actual author of d' can be derived as:

$$p\left(a, d'\right) = -\sum_{\forall ij} Q_{ij}^{a,d'} \ln\left(\frac{Q_{ij}^{a}}{Q_{i}^{a}}\right) \tag{2.52}$$

by which the given text is then assigned to the author that produces the smallest p value.

The Markov chain approaches in AA is usually operate on the character level. Khmelev and Tweedie [2002] have applied the Markov chains on the character level to three different applications with little modification to the methodology itself. The features were 26 case-insensitive characters plus the white space character. In the first experiment, 387 English texts of 45 authors were collected from Project Gutenberg. Each of these authors has more than one text in the collection. Then a total of 45 unknown texts, one of each author, were to be identified. An accuracy of 73.3% was achieved. In the second experiment, the data set used was two works of fiction that have been used by Baayen et al. [1996] initially. Five out of the six samples have been assigned with correct authorship as the best result. The last experiment was based on the 65 Federalist papers; a 9% mis-classification rate was reported. Due to the limited data, whether results of these last two experiments are meaningful is unclear.

Kukushkina et al. [2001] applied Markov chains to a collection of Russian texts. This collection consists of 385 texts of 82 authors; each was left out and identified in turn. Not only letter pairs were extracted as features, but also features of grammatical classes in Russian were used. The letter pairs were generated based on the 33 Russian characters including the space symbol. An accuracy of 73% has been achieved as the best result with letter pairs.

Peng et al. [2003a] experimented with Markov chains on the character level on collections in Greek, Chinese, and English. A number of 200 Greek documents by 10 authors have been used. For each author, documents were split equally for training and testing. Additionally, documents written by journalists were separately evaluated from scholars. The Markov chains approach was able to produce an accuracy of 74% with documents of journalists, and 90% with documents of scholars. The results also suggested that the most suitable value for N is 3. Differently, with the collection in English that consists of texts by eight authors, the best

result was obtained as 98%, given N equals to 6. The last experiment was carried out with a collection of Chinese texts from eight authors, achieving 94% effectiveness. This data was even smaller, with only one or two documents available for each of the eight authors. In this respect, the reported effectiveness may not be reliable.

Juola [1997] proposed a similar approach that can be applied to AA, in which the uni-gram model on the character level was used, and the cross-entropy from information theory was adapted. The cross entropy in the work was used to measure how likely a document was written by a certain author:

$$H = -\sum p_i \log q_i \tag{2.53}$$

where p and q are two estimated distributions, and i refers to the distinct characters. Note that for AA, the two distributions are measured from a test document d', and a set of training documents from the author a. In this respect, the calculation of the above formula is similar to the Equation 2.52; while Juola used the uni-gram model rather than the bi-gram model (first-order Markov chains). However, this work [Juola, 1997] was concerned with small corpora; the effectiveness of AA was evaluated by identifying the authorship for six disputed samples in the *Federalist Papers*. The investigation is a binary AA; six chunks from two papers, one from each author, were used as training data—each paper provides three chunks, containing 500, 1,000, and 2,000 characters respectively. The results showed that using 1000-character and 2000-character chunks resulted in a perfect assignment, while using 500-character chunks was misleading. However, due to the small data set, the goodness of this method is still unclear. In a more recent work, Juola and Baayen [2003] used the uni-gram model and cross entropy on another collection that was originally used by Baayen et al. [2002]. The corpus consists of 72 essays from 8 students (9 from each). The evaluation was based on binary AA, and the approach was used on both the character level and word level. The character-based model produced 73.2% accuracy; and, the word-based model—using 164 function words—achieved 86.9% accuracy.

A common issue in using this type of approach is the zero-occurrence problem. Some transitions may be absent in either training data or the unknown documents, which may cause an invalid natural logarithm computation. In AA literature, researchers usually define Q (or the probability) as 0 and $\ln 0 = 0$ in the cases where certain transitions are not

modelled from the training data. Additionally, in the use of Markov chains usually requires high computational complexity; it increases quadratically when the number of features is increased, and exponentially when N increases. This somewhat explains why Markov chains in AA usually operate on the character level rather than word level. For example, if bags-of-words are selected as features, the computational cost of calculating transition matrix can be tremendous. While it is possible to model the Markov chains at the word level, such as restricting to function words to reduce the number of features, there have been little results that can be used for comparison. Additionally, whether character usage is stylistic is still unclear; existing works do not provide strong evidence or proof.

2.4.4 Compression Techniques

Alternative AA techniques use compression programs to judge the similarity between pairs of data sequences. There are a variety of algorithms developed for compression; simply speaking, it is the process of encoding information using fewer bits than the original unencoded representation.

To apply compression programs for authorship attribution (AA), the document of unknown authorship is appended to a set of training samples that share authorship. Given an unknown document d_i, and a set of training samples A_j of author j, off-the-shelf compression algorithm S is then applied to the original document pool A_j, as well as the composite documents, $A_j + d_i$. The relative size after compression, ΔS, is then calculated: $S(A_j + d_i) - S(A_j)$, where $S(A_j + d_i)$ is the size of the composite data after compression, and $S(A_j)$ is the size of A_j after compression. The appended file d_i is assigned to the author j if the smallest ΔS is computed with A_j.

Benedetto et al. [2002] have applied this type of approach to different applications including AA. A standard LZ77 compression program has been used in their work. The corpus consists of 90 different texts by 11 authors, each is used as the appended file that was to be identified. In other words, to calculate the ΔS, each individual text in the corpus was appended to the 11 different document pools for each of the authors.

However this method has been criticised by Goodman [2002] who re-evaluated the method, and failed to produce promising results. Goodman also experimented with text categoriza-

Table 2.4: Results were drawn by Kukushkina et al. [2001], comparing Markov Chain to 16 compression programs.

Methods	Accuracy (%)	Methods	Accuracy (%)
7zip	47.6	Arj	56.1
Bsa	53.7	Bzip2	46.3
Compress	14.6	Dme	43.9
Gzip	61.0	Ha	57.3
Huff1	12.2	Lzari	20.7
Lzss	16.1	Ppm	26.8
Ppmd5	56.1	Rar	70.7
Rarw	86.6	rk	63.4
Markov Chain	84.1		

tion in general, and found that first, this compression-based method was 17 times slower than a naïve Bayesian approach. Second, the method produced three times more errors than a naïive Bayesian method. Third, this approach has other obvious flaws. In general, a compression program is based on modeling of character sequences, so there is a bias introduced by the subject of the text. Additionally, the method is not well designed, as it requires quadratic computational complexity. In this sense, it is intuitively not suitable for large data collections due to the low efficiency.

Kukushkina et al. [2001] also doubted the compression-based methods for AA. They have experimented with a collection of 385 documents of 82 authors in Russian. A total of sixteen popular compression techniques have been evaluated, and have been compared to a Markov chain AA approach. A brief summary of the results is shown in Table 2.4. The Markov chain based AA approach has produced better effectiveness than 15 out of the 16 compression-based results, except for "Rarw". Most of the tested compression programs have been poor, achieving less than 50% accuracy.

Compression techniques build a model of the data, then a coding technique uses the model to produce a compact representation. Typical coding techniques used in practice is to proceed at a reasonable speed, and thus may not provide a good indication of properties of

the underlying model. By using off-the-shelf compression rather than examining properties of the underlying model, much accuracy may be lost, and nothing is learnt about which aspects of the modeling are successful in AA. Therefore, whether this technique is suitable for authorship attribution is unclear.

2.4.5 Machine Learning Approaches

Machine learning approaches have been applied to AA in recent years, including neural networks [Hoorn et al., 1999; Kjell, 1994a], Bayesian classifiers [Kjell, 1994a; Coyotl-Morales et al., 2006; Uzuner and Katz, 2005], SVMs [Diederich et al., 2003; Koppel and Schler, 2004], and decision trees [Koppel and Schler, 2003]. Neural networks have been shown to be poor for AA. In contrast to neural networks, SVMs and Bayesian networks are more promising.

Diederich et al. [2003] have used SVMs[12] on a collection of newspaper articles in German. Seven authors were selected, and each of them had 82 to 118 documents written on politics and local affairs. Documents with fewer than 200 words were not used, being considered as having insufficient authorial information. The package SVM-light,[13] developed by Joachims [1998], was used. Note that there is no need to reduce the number of features in use of SVMs. Different types of features were tested, including 97,600 content words, 817 function words, 2,488 function words with corresponding POS categories (named "tagwords"), and 70,315 bi-gram tagwords. The reported overall accuracies were between 60% and 80%. The results suggested that the bi-gram tagwords were less effective amongst all types of features.

Koppel and Schler [2004] used one-class SVM proposed by Chang and Lin [2001] on a collection consisting of 21 nineteenth century English books from a total number of 10 authors. Features used in their work are a list of 250 most frequent words, not necessarily the function words. An overall accuracy of 95.7% has been achieved. However, the data set is very small, and there are only 1 or 2 books collected for each author. As we discussed before, collections of small sizes are not suitable for evaluating AA approaches, and therefore, the reported success may not be reliable.

Both SVMs and C4.5 decision tree have been applied by Koppel and Schler [2003]. A

[12]The mathematical background is described in section 2.2.4

[13]It is available at http://www.cs.cornell.edu/People/tj/svm_light

collection of 480 unedited texts of 11 authors were evaluated. These texts were emails with no processing of error checking. Three types of features have been explored: 480 function words, 59 Brill POS tags [Brill, 1992] in bi-gram form, and idiosyncratic usage (spelling errors). Both methods have been evaluated using consistent ten-fold cross validation. The results showed that using SVMs with only function words has led to an accuracy of 47.9%, while a slightly lower accuracy of 46.2% has been achieved by using only POS tags. On the other hand, the C4.5 decision tree was worse than SVMs in general, achieving only 38% and 40.4% respectively; however different from SVMs, the decision tree was more effective with POS tags rather than with function words. Combination of all features has led to much better results, in particular combining with idiosyncratic usage. The best accuracy was achieved at approximate 72% with C4.5 decision tree, using POS tags as well as the idiosyncratic usage.

In more recent work, Kaster et al. [2005] have used the SVM-light package for binary AA. The test collection used in their experiments consists of books from Project Gutenberg. A total number of 10 authors were selected, each of whom had a number of books from 7 to 129. Both lexical-based and syntax-based features were extracted, including bag-of-words, function words, POS annotation, and syntax tree. The best reported microaverged accuracy was over 90%. However, the result is not convincing, because there are many duplicates contained in the collection: some documents are copies of others, and some are segments of complete books. For instance, 89 Shakespeare works are collected in the data, however, the number of distinct works is less than a half. Books of other authors are also highly duplicated.

Additionally, Sanderson and Guenter [2006] have compared SVMs to Markov chain based approaches. Both methods have been evaluated using features at the character level. As data, texts from 50 newspaper journalists were collected with a minimum number of 10,000 words per author. Journalists writing on restrictive topics were avoided. Interestingly, SVMs have been shown to be worse than simple Markov chain based methods. The results also suggest that a number of 5,000 words were required to achieve reasonable effectiveness for AA in their experiments.

Bayesian classifiers have been competitive alternatives for many text classification tasks, including AA [Kjell and Frieder, 1992; Coyotl-Morales et al., 2006]. Kjell and Frieder [1992]

have applied both Bayesian classifiers and neural networks to identify authors of the 65 Federalist papers. N-grams at the character level have been extracted as features, where N is from 1 to 5. Both types of classifiers have achieved a reported 95% accuracy as the best result. With neural networks, short N-grams performed as well as long N-grams, however with Bayesian classifiers, short N-grams were much worse.

Coyotl-Morales et al. [2006] have carried out a study of authorship attribution of poems. The data consists of 353 poems by 5 authors. These short texts have a number of 176 words on average. A naïve Bayesian classifier with 4 types of lexical-based features has been investigated. Features were function words, content words, a combination of the previous two types, and N-grams on the word level. In contrast to the results of many other AA studies, function words have been worse than content words, achieving 41% accuracy in contrast to 73% accuracy when using content words. The results also suggested that N-grams were helpful to improve the effectiveness of AA by choosing appropriate values for N. However, higher values of N do not necessarily improve the effectiveness.

Although machine learning approaches have led to promising results in AA, none of them can be compared because none of these methods have been evaluated on the same data collections, or using the same types of style markers. The robustness and scalability of these approaches is still open to question.

2.5 Chapter Summary

In this chapter, we have reviewed current research in the areas of text categorization (TC) and authorship attribution (AA). The main focus of this thesis, that is AA, is effectively a TC task, given that it shares a general framework with TC, involving feature extraction and classification that is applied to the extracted features. However as we discussed, it differs from TC dramatically in several respects.

Despite the fact that many data sets and stylometric features have been proposed in AA, open challenges still exist, and the results are far from satisfactory. First, none of the data sets in the literature have been able to be used as a benchmark in AA. None of the collections are big enough for proper evaluations of AA techniques. Second, features cannot be directly compared due to the dramatically inconsistent experimental setup. There is no con-

sensus about which are the better features [Rudman, 1998]. Last, the proposed classification methods in AA cannot be evaluated properly.

A range of AA classification methods have been proposed; from basic statistical-based methods such as principal component analysis, Markov chains, and compression-based approaches; to machine learning methods, such as neural networks, Bayesian networks and SVMs. Results are promising, while success is subject to specific scenarios in most cases. Therefore, whether these methods are robust enough for alternative AA problems is unclear. The scalability of these approaches is a further issue, caused by the limited data that has been used for evaluations. Whether these methods can scale beyond trivial problems is still open to question.

In the next chapter, we will discuss the development of test collections that are suitable for AA. Several collections are developed for various types of AA tasks. Preliminary investigations on the newly developed data sets are also presented. Several state-of-art TC techniques, described in Section 2.2.4, are selected for the investigation.

Chapter 3

Collections and Preliminary Investigation

In Chapter 2 we reviewed the study of authorship attribution (AA) from several perspectives. One of the main challenges in AA is the lack of standard data sets that can be used for evaluation purposes. This limitation, together with diversities in the experimental design, has led to difficulties in comparing existing AA approaches in literature. It is almost the case that each researcher has their own test data that is not publicly accessible. On the other hand, most of the collections in AA are small, usually consisting of a few hundred documents at most. Although success has been claimed in previous research, the scalability and the robustness of many AA approaches are still doubtful.

This chapter focuses on the development of standard test collections for AA; these collections are designed for a variety of tasks investigated in this thesis. A preliminary investigation is also carried out with two of our newly-developed collections in this chapter—named *AP7* and *APoc*—aiming to establish the value of using a benchmark in AA, and to further explore whether the existing TC techniques can be used for effective and scalable AA.*

*This chapter incorporates work originally published by Zhao and Zobel [2005].

3.1 Developing Good Test Collections

In contrast to general text categorization (TC) that is concerned with the content or topics of a given text, authorship attribution (AA) is concerned with the author or writing style of that text. Since, these two applications aim to address the classification in different respects, benchmarks developed for TC are not guaranteed to be suitable for AA.

An appropriate test collection for AA should meet several basic criteria:

- The collection should be able to provide a sufficient number of documents for training; each of them should have identified or valid authorship. Small volumes of text are usually not sufficient to reflect authors' writing styles. Here, identified or valid authorship refers to the name of a person, not any form of role or organization. Also, the documents selected should not be co-authored.

- The collection should be reasonably large, not too small, so that the scalability of proposed AA techniques can be properly evaluated.

- Documents in the collection should not be written on restrictive subjects, following the assumption introduced in Chapter 2 that the writing style of a particular author is believed to be independent of topics. Therefore a robust AA technique should be able to effectively identify authors of documents written on various topics.

- The collection should be possible to carry out one or more types of AA experiments with the collection, including binary AA, multi-class AA, and one-class AA.

By considering the above criteria, several data collections are developed for authorship attribution in this thesis. The documents in collections are drawn from two domains: newspaper articles from *The Associated Press*, and English literature from *Project Gutenberg*.

The Associated Press (AP) is a sub-collection of newswire articles from the TREC corpus [Harman, 1995]. Seven collections of newswire articles are developed using AP data (Full details of these collections are given in Section 3.1.1). Two of them, *AP7* and *APoc*, are used for our baseline experiments in this chapter. Briefly speaking, *AP7* is a collection designed for binary AA and multi-class AA; it is the main collection that we use to evaluate our entropy-based AA technique in Chapter 4 and Chapter 5. *APoc* is designed for one-class AA. Five

other collections, *AP10k*, *AP100k*, *AP500k*, *APvote10k*, and *APvote100k*, are much larger collections, which are developed for evaluation of authorship search in Chapter 6.

Project Gutenberg is a publicly accessible website that provides a great number of e-books for free.[1] *GutenbergSmall* and *Gutenberg634* are two collections of classic literature in English; the first is a segment-based collection, and the second is a book-based collection (Full details of these collections are given in Section 3.1.2). Neither of these collections contain duplicate texts. *GutenbergSmall* is used in Chapter 4, evaluating the new entropy-based AA method for comparable results; the *Gutenberg634* collection is used for evaluating authorship search methods in Chapter 6.

3.1.1 The Associated Press

The AP collection consists of more than 250,000 documents that have been written by more than 2,380 distinct authors over several years. Approximately 60% of the documents in the collection have valid authorship, while the others are anonymous or have invalid authorship. In this context, to be valid, the authorial information of a document should refer to a single person.

We believe that AP can be used to develop standard collections to evaluate AA techniques for several reasons. First, AP is large so that the scalability of an AA method can be examined in two different ways: by increasing the number of documents, and by increasing the number of authors. There are many authors who regularly contribute articles to AP; the biggest number of documents written by a particular author is more than 800, after removing duplicates; more than 10% of the authors have contributed over 100 texts to AP. This indicates that there should be good volume of evidence for these authors to some extent. In previous AA research, the largest collection is that used by Diederich et al. [2003], in which 100 documents on average were used per author. In this respect, AP is able to provide enough documents and enough authors for AA investigation; evaluations can be carried out on both the effectiveness and scalability.

Second, documents in AP cover a wide range of topics, with some authors contributing diverse material while others are specialized. This poses a further challenge for effective

[1] http://www.gutenberg.org

authorship attribution (AA); the robustness of an AA method can be properly examined.

Third, the documents in AP have been edited for publication, meaning that they are largely free of errors, if not absolutely. However, drawbacks still exist within the AP collection itself with instances of: multiple versions of the same document; multiple versions of names of the same author; and, potential typographic errors in presenting author names. In order to address these issues for developing good test beds for AA, we apply an error-pruning process as described in the following sections.

Eliminating Near-duplicate Documents

It is often the case that AP contains multiple versions of the same document in the collection. This is because the same article may be published in slightly different forms in different places and newspapers. This kind of repetition can distort the statistics used to evaluate attribution, leading to overestimates of the attribution accuracy. For instance, a nearest-neighbor approach (as described in Chapter 2) will automatically be successful if the test document is also presented in the training data. However elimination of these duplicates is not trivial due to the fact that these kind of documents are not direct copies.

We use the terminology *near-duplicates* for these documents of multiple versions. The *SPEX* method proposed by Bernstein and Zobel [2004] is applied to discover the near-duplicate documents in the AP collection. The number of duplicate versions of a particular document can vary significantly. For instance, the document *AP880906-0189* has 23 duplicates; both *AP900320-0172* and *AP890531-0246* have 19 duplicates; while most of the discovered documents have only 1 extra version. 3,303 documents have been duplicated; a total of 3,719 duplicates are eliminated by this process.

Standardising Inconsistencies in Authorship

Another issue is caused by the inconsistencies of authorship presentations in AP [D'Souza et al., 2004]. For the same author, there are multiple versions of the author names presented in the collection, especially of those authors who have middle names. The middle name of a particular author is sometimes omitted, sometimes written in shortcuts, and sometimes fully expanded. One of the possible reasons may be that different publishers have different

Table 3.1: Examples of possible versions of author representations in AP. N_d is the number of documents written by author A; N'_d is the number of documents written by author A', where A and A' may refer to the same writer. The value of R is the ratio between N_d and $N_{d'}$. Differences in the names are in italic font format.

Author (A)	N_d	Author (A')	N'_d	R (N'_d/N_d)
Abir Taha	1	Abir *Riad* Taha	1	1
Alberto Franco	1	Alberto *S* Franco	20	20
Anita Huslin	1	Anita *C* Huslin	10	10
AV Gallagher	1	*A V* Gallagher	28	28
Charlene Fu	3	Charlene *L* Fu	103	34

Table 3.2: Examples of possible typographic errors in names in AP. N_d is the number of documents written by author A; N'_d is the number of documents written by author A', where A and A' may refer to the same writer. The value of R is the ratio between N_d and $N_{d'}$. Differences in the names are in italic font format.

Author (A)	N_d	Author (A')	N'_d	R (N'_d/N_d)
Abd*e*l Jalil Mustafa	11	Abd*u*l Jalil Mustafa	3	0.27
Abebe Andual*e*m	16	Abebe Andual*a*m	2	0.13
Ahm*e*d Mantash	40	Ahm*a*d Mantash	3	0.08
Alina Guer*r*ero	53	Alina Guer*e*rro	1	0.02
Andrew Ka*r*ell	396	Andrew Ka*t*ell	1	$3e^{-4}$

templates for authors to follow. A few examples are provided in Table 3.1; for instance, the authorship *Abir Taha* and *Abir Riad Taha* may or may not refer to the same author.

In addition to the middle names, there are potential typographic errors in representations of author names in the collection. Names that differ in only one or two characters are more likely typographic errors rather than different authorship. We present some examples in Table 3.2. The notations used in this table have the same meanings to those used in Table 3.1. However, there is no foolproof way for us to judge whether the different representations actually indicate the same writer. Thus, we make decisions about the name variants of a

Table 3.3: Statistics of the AP7 collection. N_d refers to the number of documents. L_{min} is the length of the shortest document written by a particular author; L_{max} is the length of the longest document; and, L_{av} is the average length of all documents of that author.

	Author						
Statistics	Schweid	Currier	Skidmore	Dishneau	Kendall	Crutsinger	Beamish
N_d	941	843	965	818	1001	801	894
L_{min}	49	15	136	253	28	224	201
L_{max}	1880	7810	1645	8393	4943	1829	2569
L_{av}	632	517	596	644	705	674	651

certain author, and standardise the authorship. In both cases, if $R \leq 0.05$ or $R \geq 20$ then we consider author A and A' as identical, where the ratio R is measured by N'_d/N_d as shown in the tables; texts of a particular authorship class are then combined for use.

While the collection may not be absolutely error free after applying the above methods, it is more appropriate for developing standard AA collections, and we do not have the evidence necessary to make an error-free collection. Base on the processed AP data, several collections are developed.

AP7 and APoc. We select seven authors who are regular contributors in AP.[2] Documents written by these seven authors are compiled to form the *AP7* collection. Each of the authors has over 800 documents available in AP after error pruning; Table 3.3 shows some statistics on this collection.

In the table, N_d is the number of documents written by each of the seven authors; L_{min}, L_{max}, and L_{av} are respectively the minimum, maximum, and the average document length for that author. These basic statistics are based on individual words; digits and punctuation symbols are not considered. The average length of a document in the collection is 724 words.

As shown, the average length of documents differs greatly amongst the seven authors. For instance, Kendall writes much longer newswire articles than does Currier. This motivates

[2]The selected authors are: Barry Schweid, Chet Currier, Dave Skidmore, David Dishneau, Don Kendall, Martin Crutsinger, and Rita Beamish.

us to examine the distributions of document length of each author. We note that even with this surface property of the documents, different authors tend to have different preference. The distributions of document length are depicted in Figure 3.1. Clear differences can be observed; for instance, the distribution in relation to Beamish is more bell shaped compared to the other authors, while the distribution of Kendall has a heavier tail. A document of approximately 500 words long is more likely to be written by Currier rather than by any of the other six authors. This supports to a slight extent that authors do have different writing habits.

The AP7 collection can be used for both binary authorship attribution and multi-class authorship attribution of up to seven classes. AP7 is intuitively a harder collection compared to many data sets that have been used in prior research, as discussed in Chapter 2. First, the articles are generally short in AP7, only 724 words on average, while the numbers are usually several thousands or more in previous research. Additionally, the articles may be written to a template or house style of a particular publisher, meaning that the writing maybe changed to some extent from the original. In contrast to material drawn from sources such as literature, we would not expect human readers to be aware of strong stylistic differences between the authors in AP. The AP7 collection is the main data used in Chapters 3, 4, and 5.

The APoc data, another collection developed from AP, consists of a further 10,000 anonymous documents. It is designed for one-class AA, in which a large number of negative samples are essential as introduced in Chapter 2. We include 10,000 negative samples in APoc, so that the scalability AA approaches can be investigated. This collection has been explored for the preliminary investigation as well in this chapter.

AP10k, AP100k, and AP500k. These collections are developed to investigate authorship search that can scale to much larger collections. From the AP7 data, we randomly select 700 documents, 100 for each of the seven authors. These 700 documents are consistently included in all the three collections. To form the *AP10k* collection, which consists of 10,700 documents in total, a further 10,000 anonymous documents from AP are included. In addition, *AP100k* consists of 100,700 documents in total, which is a superset of the AP10k collection. The *AP500k* collection consists of documents from AP, WSJ, and SJM collections;

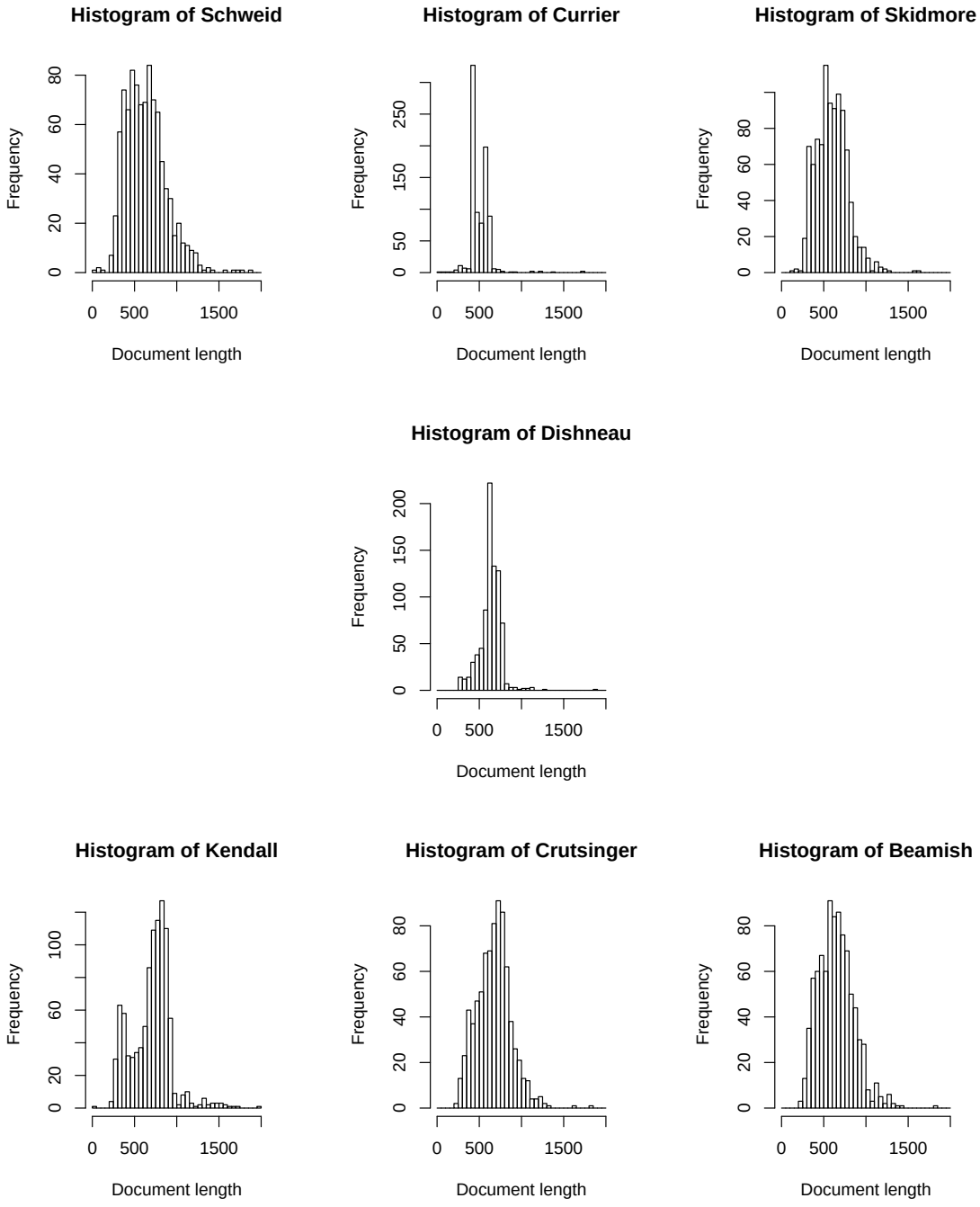


Figure 3.1: The distribution of document length in words for different authors.

all texts are newspaper articles. A total of 500,700 documents are in this collection.

APvote10k and APvote100k. In contrast to the AP10k collection, documents in both *APvote10k* and *APvote100k* collections have a further requirement that all documents have to carry valid authorship information. In APvote10k, there are 10,000 documents from 342 distinct authors, while the APvote100k collection consists of 100,000 documents by 2,229 authors. The same set of 700 documents from the consistent seven authors in the AP10k are included. Also, over 10% of authors in both collections have more than 100 contributions.

The first three collections are developed for investigations of authorship search (introduced in Chapter 6), while the APvote10k and APvote100k collections are used to evaluate search-based AA approaches for large document collections. All five collections are carefully investigated in Chapter 6.

3.1.2 Project Gutenberg

A key aspect of AA investigation is to explore the effectiveness of attribution on literature. Compared to newswire articles, literature is considered to have stronger stylometry [Kaster et al., 2005]. In some early studies of AA, section-based or segment-based collections have been explored [Baayen et al., 1996; Khmelev and Tweedie, 2002].

Project Gutenberg has over 19,000 free e-books available for public access. This includes not only materials in English but documents in other languages as well; we only consider English materials in this thesis.

There is no strict template for volunteers to distribute texts to Project Gutenberg website. The format of books differs significantly; materials can be distributed in plain text, in HTML format, or even in MP3 audio format. We choose to download books in plain text format from the website; text in each downloaded book needs to be cleaned carefully to make it appropriate for AA.

Each individual e-book contains some style-free materials that are not written by the author of that book. Most of them have a consistent opening, followed by the *Project Gutenberg license*. These texts are tedious, and the content is edited inconsistently by different volunteers. For example, some books start with a fairly short opening, of only a few lines,

while some books have over a thousand lines as the opening. The style-free information is also included at the end of each book. Similarly, some books have only a few words as the closing text, while others have several hundreds of lines at the end. In addition, sometimes some volume of text in a book is not written by the author of that book, such as the preface, which should also be cleaned. However, elimination of these materials is not straightforward, as there are no such symbols that we can use to locate these types of texts correctly in each book individually; the occurrences and presentations are different from book to book. Take the closing materials for instance: it is simply "— End —" in some books; "The end of gutenberg" and "END OF GUTENBERG". In addition, there are other variations: edited in either lower case or upper case, using special symbols between characters such as - and *. To create proper test beds based on the downloaded books, we manually clean the style-free materials, that is, separating them from the real texts of books. Two collections are drawn for AA evaluations; neither of them contain duplicates.

GutenbergSmall. This collection consists of segment-based or section-based texts. Here, a segment or a section refers to a volume of text extracted from a complete document; examples are: a chapter of a book, a paragraph of an article, and a number of lines of a complete program code. A total of 137 books written by five well-known authors[3] are considered.A chapter is selected as a segment; each book is therefore broken into individual chapters. Each of the five authors have a number of chapters ranging from 492 to 1,174. The average length of a segment is 3,177 words, much longer than articles in the AP data. Like the AP7 collection, this corpus is also used in Chapter 4 to evaluate AA methods.

Gutenberg634. We collect literature that was representative and consistent from Project Gutenberg. We gather books from 55 of the top-100 most popular authors, that is, books of those authors that are downloaded the most. For most authors, we download 10 books, or fewer if less than 10 books are available; for some authors we collect all works.

The total number of books collected is 634, which are by 55 authors; the collection is named *Gutenberg634*. In selecting the books, we do not collect volumes of poetry, dictionaries, and texts in languages other than English; short stories are avoided as well. In addition,

[3]Authors are: H. Rider Haggard, Thomas Hardy, Leo Tolstoy, Anthony Trollope, and Mark Twain.

authors with four or fewer books are not considered. We keep both novels and plays; plays are greatly in the minority. These plays are collected from *Shakespeare* and his contemporaries.[4] More details are provided in Chapter 7.

3.2 Preliminary Investigation

We undertake a preliminary investigation of AA using two of the developed collections: AP7 and APoc. AP7 is used to evaluate binary AA and multi-class AA, while APoc is used for one-class AA. Our aim in this investigation is to examine whether the developed collections can be used as standard benchmarks to evaluate the relative performance of different attribution methods. We make use of the successful TC classifiers for AA on the AP7 collection; both binary AA and multi-class AA are investigated. One-class AA is evaluated on the APoc collection. A set of 363 function words[5] are used as features, and results are reported for several techniques, including well-known machine learning text classifiers.

3.2.1 Stylometric Features

As discussed in Chapter 2, a wide range of different style markers have been proposed in prior research in the field of AA, from the token-based features such as the simple document length, to advanced syntax-based features such as the syntax tree. Documents can be seen as combinations of words; a straightforward choice is to use words in documents as the features. However, content words can be misleading in AA—as we show later in this thesis—it is therefore interesting to restrict attention to function words. These are words such as prepositions, conjunctions, or articles, or elements such as words describing quantities, that have little semantic content of their own and usually indicate a grammatical relationship or generic property.

The appeal of function words is that they can be a marker of writing style. Some less common function words—such as *whilst* or *notwithstanding*—are not widely used, and thus may be an indicator of authorship. On the other hand, even common function words may

[4]The playwrights are: William Shakespeare, Ben Jonson, Christopher Marlowe, and Francis Beaumont & John Fletcher (whose works are co-authored).

[5]A complete list of selected function words is provided in Appendix A.

Table 3.4: An example of usage statistics for common function words for two authors. Each number is, for that author, the percentage of function word occurrences that is the particular function word. Counts are averaged across a large set of documents by each author.

	a	and	for	in	is	of	that	the
Barry Schweid	6.28	9.22	4.94	6.50	1.62	14.66	1.89	29.13
Don Kendall	9.75	7.08	2.36	7.98	3.05	13.16	5.73	41.29

be used to distinguish between authors. Table 3.4 gives an example of how usage of function words can vary. In this example from the AP7 data, both authors use *and* and *of* with similar frequency, but Schweid's usage of *that* is a third of Kendalls's, and even the usage of *the* is very different.

Function words have been one of the most effective features in previous AA studies [Binongo, 2003; Burrows, 1987; Holmes, 1994; Holmes et al., 2001; Juola and Baayen, 2003; Pol, 2005]. However, there is no consensus on the function words; we collect a list of 363 function words to form a pre-defined feature set. This feature set is provided in Appendix A, and is consistently used as one of the marker types throughout this thesis.

Collections, such as AP7 and APoc, might be regarded as relatively challenging for the task of authorship attribution, as articles with different authors may be edited towards a corporate standard and an author may use different styles for different kinds of article; for example, some authors write both features and reviews. Furthermore, the texts are usually much shorter than literature texts.

3.2.2 First Try: Principal Component Analysis

Many studies in AA have applied principal component analysis (PCA)[6] and have reported success, in particular for binary AA tasks [Baayen et al., 1996; 2002; Binongo, 2003; Burrows, 1992; 2002; Holmes et al., 2001]. However in the study conducted by Hoover [2001], the scalability of PCA has been reported as poor for large number of documents and for more than two authors.

[6]Details of the theory are discussed in Chapter 2. Simply speaking, after PCA, the most significant components are used for discrimination, usually the first two.

We first apply PCA to the AP7 collection. For each of the seven authors, the numbers of documents chosen for sampling are varied from 20 up to 600. The pre-defined 363 function words are used as features being extracted; PCA is then applied to the extracted features. The analysis is undertaken from two different perspectives: how PCA scales when increasing the number of documents per author for sampling, and how PCA scales when increasing the number of authors. Similar trends are observed to those reported by Hoover [2001].

We start with the smallest number, that is 20 documents per author; to some extent, this number is comparable to many early AA studies. For instance, Holmes et al. [2001] used a collection of 17 journal articles written by an author for analysis; Binongo [2003] used a total number of 15 Oz books; and Hoover [2001] used 50 sample documents in total in the first experiment, where each author has only one or two samples. We then increase the number of sampling documents per author, in steps of six, up to 600.

For illustration, we use two authors: Schweid and Currier. Results are shown in Figure 3.2, plotted as the first principal component (PCA1) against the second principal component (PCA2) after the analysis; other components are discarded in agree with many prior PCA-based research. The top graph depicts the result of applying PCA using 20 documents per author; 100 documents per author are used for the graph in the bottom of Figure 3.2. We use "1" to refer to author Schweid and "2" for Currier for visual clarity of the graphs. As observed, PCA can clearly separate documents written by two different authors, given small samples. However, when the collection gets bigger, the effectiveness of PCA degrades; larger proportion of overlapping is observed from the figure using 100 texts than that using only 20 texts. When further increasing the number of documents to 600, the results show clear failures. Consistent results are obtained for all 21 pairs of authors.

Next, we increase the number of authors for attribution. PCA fails to separate documents of different authors; an example is shown in Figure 3.3. In this example, the same documents of Schweid and Currier are used as those used in Figure 3.2; further sets of documents by Skidmore and Dishneau are included for three class and four class attribution. We use notations "3" and "4" to represent documents of these two authors. Only 20 documents per author are used, since PCA fails to handle large number of documents even with easier binary attribution. The results show that, when using the most two significant principal

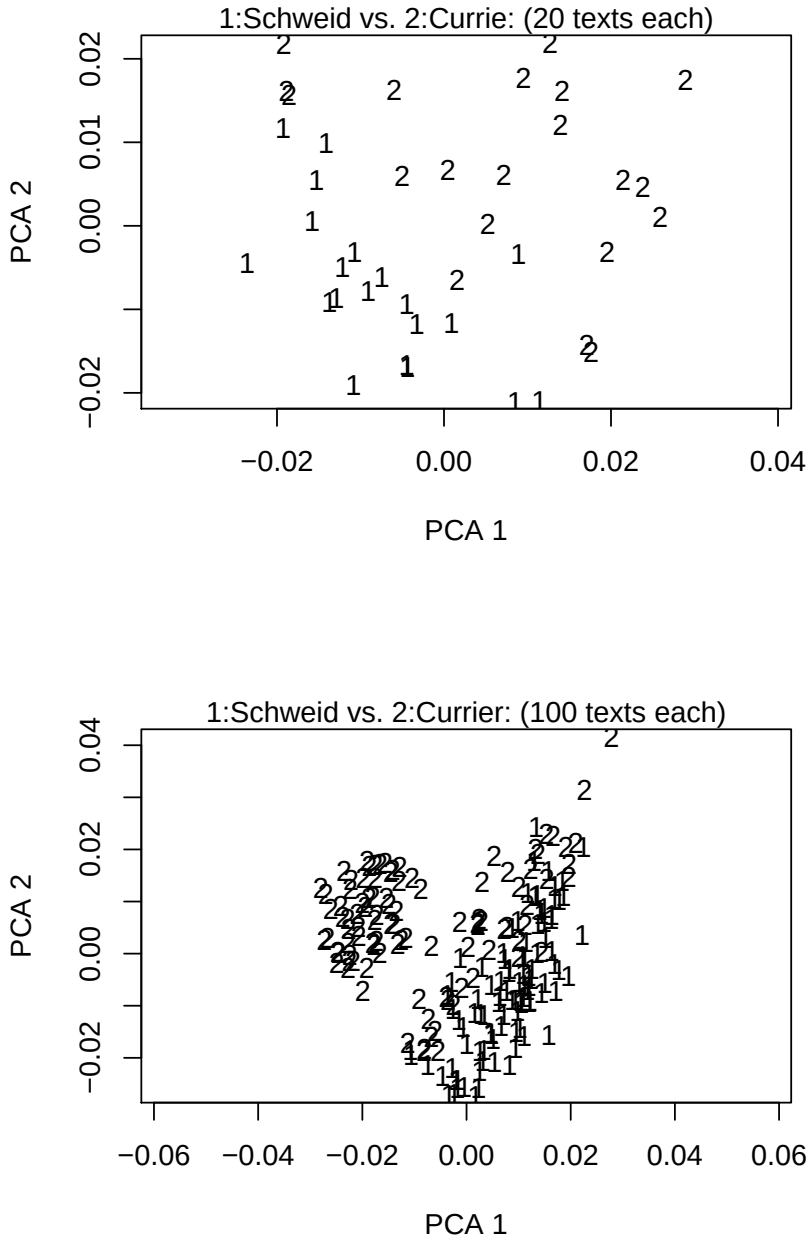


Figure 3.2: Examples of applying PCA to binary AA. 20 sample texts are available for both authors in the top figure, while 100 texts are included in the bottom figure (For visual clarity, we use "1" to represent author Schweid and "2" for Currier).

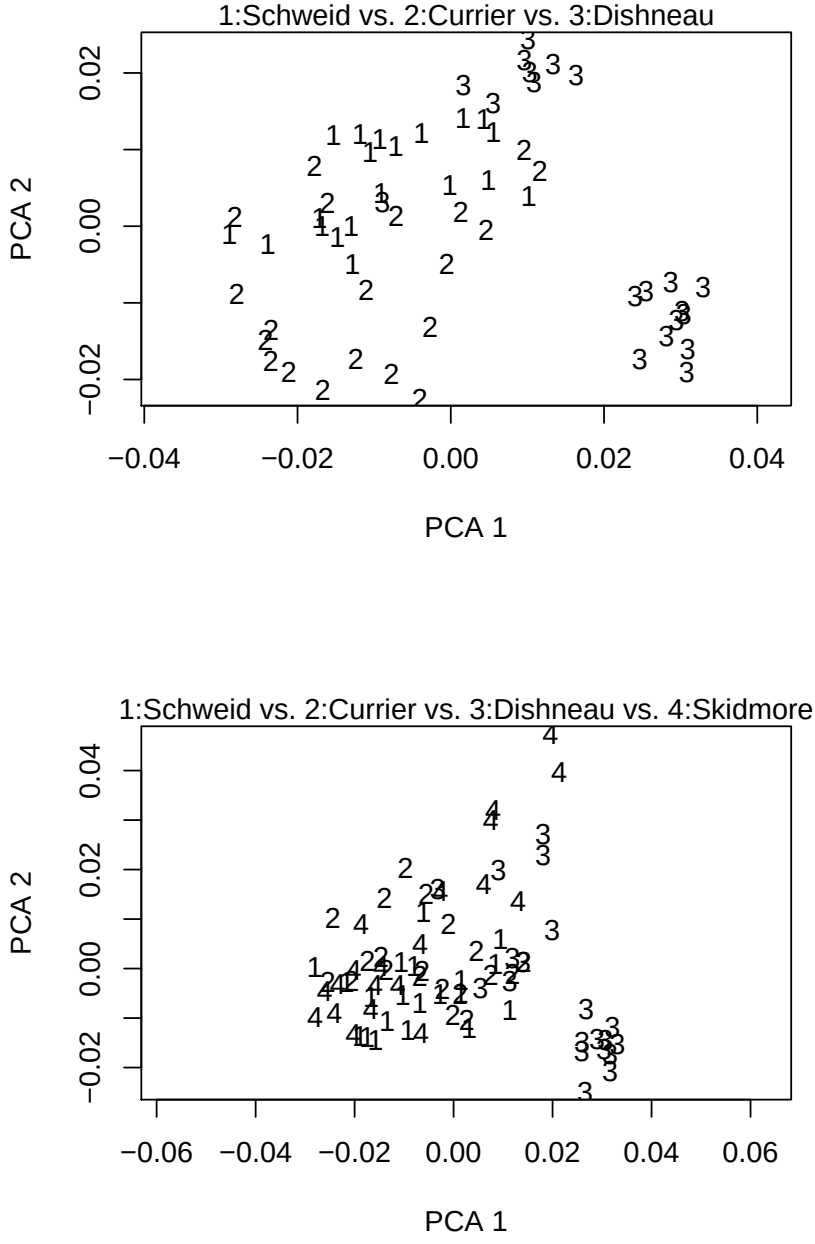


Figure 3.3: Examples of applying PCA to differentiate documents of 3 authors; the number of sample documents is 20 (For visual clarity, we use "1" to represent author Schweid, "2" for Currier, "3" for Dishneau, and "4" for Skidmore).

components in this example, PCA is still able to separate documents amongst three authors to some extent, while with 4 authors, it clearly fails. If more documents for each author are included, 50 for instance, PCA shows even worse performance; it fails even with three authors. It may be true that using a few more components would somewhat improve the effectiveness of this approach; however, there are challenges. From a presentation point of view, it is not feasible to represent the clustering using more than two or three components, that is, in a high-dimensional space; and, the results can be difficult to interpret.

We believe that PCA is neither scalable nor effective for AA in general, in particular in cases where relatively large document collections or more than two author candidates are involved. Therefore, we suggest that PCA is not a suitable technique to attribute documents in the AP7 collection, not even for binary AA tasks.

3.2.3 Baseline Experiments

Technically, AA follows a general framework of text categorization (TC), where many current techniques can scale to large volumes of data across a variety of topic classes. Intuitively, exploring the existing TC techniques for AA purposes is worth investigating.

In Chapter 2, we introduced some successful machine learning approaches for text classification; six of them are selected for our baseline experiments on the two collections. The first two are Bayesian classifiers [John and Langley, 1995; Langley and Sage, 1999]. There are several variations of Bayesian classifiers. Among them, naïve Bayesian and Bayesian network classifiers are reported as successful and have been applied to document classification [Sebastiani, 2002]. The next two, nearest-neighbor and k-nearest-neighbor, are distance-based methods; such methods compute the distance between a new item and existing items in different classes to make a decision. These two instance-based approaches are also known as lazy learning methods due to the fact that there is no model learnt during the training stage. C4.5 is a decision tree algorithm, and reputedly one of the best algorithms in the decision tree family. The last technique is support vector machines (SVMs) that is regarded as one of the best classifiers. However it is not straightforward to apply SVMs to multi-class classification, and therefore, only binary AA is evaluated and compared using SVMs in this preliminary investigation.

The two Bayesian approaches are based on the probability theory. The nearest-neighbor methods measure vector differences. Decision trees are based on classifying training data by their distinguishing features. SVMs aim to find the optimal hyperplane that separates instances of one class from the other. The first five classifiers are from the public domain—theWEKA[7]toolkit [Witten and Frank, 2000], and the SVM package used is SVM^{light}.[8] In the following, we show the results of investigating the use of classification with function words as features, using consistent document collections and varying the numbers of documents.

We use the classification methods in a variety of ways, to examine their robustness and their ability to scale. Previous research usually uses the attribution methods for binary AA, that is, to discriminate between two known authors. In this context, all the documents used for training and testing are written by these two candidates. There is a natural generalization to n-class AA for any $n \geq 2$. One-class AA is used to determine whether the given text was written by a particular author. In contrast to the n-class problem, the negative examples do not have to be by particular authors; they can be anonymous or by any other author. Cross validation is an approach designed for evaluation purposes when the amount of data is limited. In our experiments we use the standard 10-fold cross validation, where the data is split into a fixed number of ten sets of similar size. Each fold in turn is classified, while the remaining folds are used for training.

Holding the number of folds to a fixed number means that results are obtained in a consistent way, but also means that results at different scales may not be comparable, as both the test and training data has changed. For this reason, in other experiments we reserve small sets of documents as test data, while varying the number of positive and negative documents used for training to make the results directly comparable.

To establish which attribution method is in practice the most effective—and to further demonstrate the value of a benchmark—we examine how well each of the methods scales. Scaling has many aspects: increasing the volume of positive training data, increasing the number of authors, and increasing the volume of negative training data. This last two cases are of particular interest in a domain such as a newswire, where the number of documents

[7]Package is available at http://www.cs.waikato.ac.nz/ml/weka.

[8]Package can be fetched at http://svmlight.joachims.org/.

Table 3.5: Effectiveness (percentage of test documents correctly attributed) of each method for attribution, using 10-fold cross-validation on two-class classification.

# of documents per author	NaïveBayes	BayesNet	NN	3-NN	C4.5
25	81.2	81.4	81.0	80.2	69.5
50	85.1	86.0	85.5	84.6	77.1
100	85.9	89.7	83.4	82.9	80.3
200	85.8	89.3	84.3	84.1	82.9
400	85.6	90.1	85.3	85.6	84.8
600	85.5	90.5	85.8	85.5	84.5

and authors is large.

Binary Authorship Attribution: With Weka

In the first experiment, we compare the five classification methods from the Weka package, using 10-fold cross-validation and two-class classification on AP7 collection. These results are directly comparable. We vary the size of the total document pool to see how the methods behave at different scales. There are 7 authors in the collection; therefore the experiments are carried out with a total number of $21 = C_7^2$ possible author-pair combinations. Results are shown in Table 3.5, where outcomes are averaged across all 21 pairs of authors. Several trends can be observed. The first, and perhaps the most important, is that function words can indeed be reliably used for authorship attribution.

All the methods become more effective as further documents are included, but only up to a point; only for the decision tree does effectiveness significantly improve for classes of more than 100 documents. For larger sets of documents, little separates four of the methods, but Bayesian networks are markedly superior.

In the second experiment, we randomly select 100 documents per author and set them aside consistently as for testing; the training samples are then extracted from the remaining document pool with varying sizes. Unlike in the previous experiment using 10-fold cross validation, where both training and testing samples are changing concurrently, in this experiment, the testing documents are set as a constant. Therefore the results are more comparable,

Table 3.6: Effectiveness (percentage of test documents correctly attributed) of each method for attribution, using the same 100 test queries per author on two-class classification. Results are averaged across eleven pairs of authors.

# of documents per author	NaïveBayes	BayesNet	NN	3-NN	C4.5
25	68.8	79.9	67.2	69.2	62.1
50	78.9	82.0	75.7	77.9	73.6
100	81.6	85.7	76.3	78.3	79.0
200	84.2	88.2	80.0	81.5	82.6
400	84.8	90.6	80.0	80.9	86.2
600	84.5	90.6	80.7	81.5	86.7

since the effectiveness of different methods is evaluated based on the attribution of the same set of test documents. We run the experiment on each of the 21 author pairs, and the reported results are an average across these runs. These results are shown in Table 3.6; as shown, the methods are more clearly separated than was the case in the first experiment; the nearest-neighbor methods are poor, while Bayesian networks are more effective at all scales, with slightly increasing accuracy as more training documents are included. Also, we observed significant inconsistency from one pair of authors to another, throwing considerable doubt over the results reported in many of the previous papers on this topic, most of which used only two authors.

Table 3.7 presents the effectiveness of binary AA on an author-by-author basis, given 25 and 200 training samples per author, respectively. The numbers in bold are the highest accuracies achieved by the methods in relation to individual binary AA tasks. There are 11 binary AA tasks for each case, provided for illustration. From Table 3.6, we note that given small training samples—25 documents per author—the Bayesian networks are more effective than other methods, by 11% at least; given 200 documents per author for training, the overall differences are smaller. However, the Bayesian networks are not always better than other methods for individual attribution tasks, as indicated in Table 3.7. For instance, with 200 training documents, the C4.5 decision tree algorithm is very poor for the task with ID 6, giving only 59% accuracy even worse than a random attribution. However it performs the

Table 3.7: Effectiveness (percentage of test documents correctly attributed) of each method for attribution, using the same 100 test queries per author on two-class classification. Results are for individual pairs of authors, using respective 25 and 200 training samples per author. The highest accuracy in each case is in **bold**.

# of training	ID of pairs	NaïveBayes	BayesNet	NN	3-NN	C4.5
25 samples						
	1	88.0	**97.0**	84.0	80.0	69.5
	2	**85.0**	**85.0**	79.5	71.0	83.5
	3	**82.0**	81.5	64.0	79.0	71.0
	4	**92.5**	80.5	78.0	86.0	58.0
	5	68.5	59.0	66.5	**77.0**	62.0
	6	59.0	**59.5**	53.5	57.5	56.5
	7	80.0	**90.0**	89.0	90.0	76.0
	8	82.5	**93.0**	80.0	81.0	70.0
	9	88.5	**89.0**	76.0	78.0	73.0
	10	**84.0**	82.0	81.5	82.0	69.5
	11	82.3	**93.0**	77.0	80.5	92.0
200 samples						
	1	93.5	95.0	85.5	82.5	**96.0**
	2	**91.0**	88.0	85.0	83.5	**91.0**
	3	86.0	86.0	85.5	**88.0**	83.0
	4	76.0	**91.5**	78.5	82.0	68.0
	5	79.0	**89.0**	74.0	83.0	79.0
	6	71.5	**74.0**	63.5	61.0	59.0
	7	86.0	94.0	92.0	90.5	**96.5**
	8	85.5	**96.0**	80.0	81.5	92.5
	9	90.0	94.0	79.0	79.0	**96.0**
	10	83.5	**93.0**	77.0	77.0	**93.0**
	11	92.5	**97.0**	82.0	84.0	95.5

Table 3.8: The student t-test between different pairs of methods with binary AA, at a significance level of 0.05. The value of $|difference|$ is the numerical difference between each pair of methods.

# of samples	method 1	method 2	$\|difference\|$	p-value	significant?
25 samples					
	NaïveBayes	BayesNet	11.1%	0.027	Yes
	NaïveBayes	NN	1.6%	0.051	No
	NaïveBayes	3-NN	0.4%	0.045	Yes
	NaïveBayes	C4.5	6.7%	0.070	No
	BayesNet	NN	12.7%	0.011	Yes
	BayesNet	3-NN	10.7%	0.195	No
	BayesNet	C4.5	17.8%	0.003	Yes
	NN	3-NN	2.0%	0.157	No
	NN	C4.5	5.1%	0.205	No
	3-NN	C4.5	7.1%	0.064	No
200 samples					
	NaïveBayes	BayesNet	4.0%	0.006	Yes
	NaïveBayes	NN	4.2%	0.014	Yes
	NaïveBayes	3-NN	2.7%	0.085	No
	NaïveBayes	C4.5	1.6%	0.541	No
	BayesNet	NN	8.2%	$1.664e^{-4}$	Yes
	BayesNet	3-NN	6.7%	$2.676e^{-4}$	Yes
	BayesNet	C4.5	5.6%	0.117	No
	NN	3-NN	1.5%	0.398	No
	NN	C4.5	2.6%	0.046	Yes
	3-NN	C4.5	1.1%	0.116	No

Table 3.9: Effectiveness (percentage of test documents correctly attributed) of SVMs for attribution, on two-class classification. The results are compared with the Bayesian networks shown in previous experiments. The student t-test is carried out with multiple sets of training data, from 25 samples to 600. The significance level is set as 0.05.

	Method	25	50	100	200	400	600
10-fold CV	SVM	81.0	87.2	89.4	91.1	92.0	92.6
	BayesNet	81.4	86.0	89.7	89.3	90.1	90.5
	Significant?	No	No	No	Yes	Yes	Yes
Train-Testing	SVM	80.1	85.8	89.3	91.1	92.4	92.9
	BayesNet	79.9	82.0	85.7	88.2	90.6	90.6
	Significant?	No	No	Yes	Yes	Yes	Yes

best for task with ID 1, 7, 9, and 10. Thus, it is intuitively unsound to conclude a decision tree is either effective or inferior for AA based on a single test. In the literature reviewed in Chapter 2, many successful results are reported based on any two authors.

To examine whether the overall differences in effectiveness are statistically significant, we undertake a paired student t-test on the effectiveness shown in Table 3.7. The results are presented in Table 3.8, grouped by the numbers of training samples. The results suggest that, when the corpus is small, even big numerical differences may not be statistically significant; for instance, given 25 training samples, the effectiveness of Bayesian networks is 10.7% higher than the nearest-neighbour network, however the difference is not statistically significant, while the 0.4% difference between naïve Bayesian and 3-nearest-neighbour is statistically significant. On the other hand, we also note that, when having a larger corpus, the result trends are more stable; the bigger numerical differences in effectiveness, the more likely it is that the compared methods are also significantly different.

Binary Authorship Attribution: With SVMs

SVMs are fairly competitive methods for classification that have been proposed in more recent years. As one of the state-of-art machine learning classifiers for binary classification, SVMs are also evaluated under the same experimental design. Table 3.9 shows the results; we also

Table 3.10: Effectiveness (percentage of test documents correctly attributed) of each method for attribution, using 10-fold cross-validation on two to five class classification.

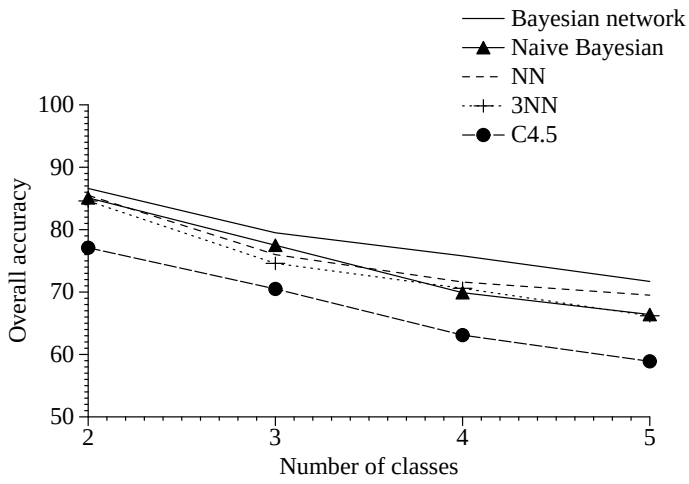
Number of classes	NaïveBayes	BayesNet	NN	3-NN	C4.5
50 documents per author					
2	85.1	86.0	85.5	84.6	77.1
3	77.5	79.5	76.0	74.6	70.5
4	69.9	75.8	71.6	70.6	63.1
5	66.4	71.7	69.5	66.2	58.9
400 documents per author					
2	85.6	90.1	85.3	85.6	84.8
3	76.5	85.2	78.7	79.0	75.0
4	70.5	80.6	73.7	74.0	67.2
5	66.0	76.3	70.5	67.0	62.2

list the best achievable results from the previous shown. SVMs outperform the other five machine learning classifiers in most cases, in both types of evaluations. As discussed in Chapter 2, SVMs usually require large amounts of training data as well as features, and thus, it is not clear whether the effectiveness obtained with small sets of training data is reliable. Also, our significance test suggests that when the training set is small, the differences are not statistically significant.

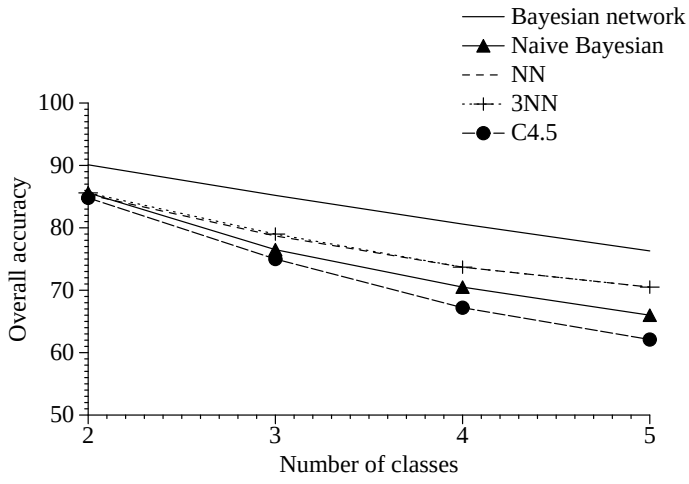
As we reviewed in Chapter 2, SVMs are usually applied to binary classification, thus, we do not use SVMs in the later experiments.

Multi-class Authorship Attribution

In the next experiment we increase the number of authors, examining the effectiveness as the number is increased from two to five. Results are averages across different sets of authors: we use 21 combinations of two and of five authors, and 35 combinations of three and of four authors. Results, shown in Table 3.10, are for 10-fold cross validation. The top half of the table is with 50 documents per author, with 400 per author in the bottom half. With approximately 400 training documents, all selected methods reach the plateau for binary AA.



Given 50 documents per author for training



Given 400 documents per author for training

Figure 3.4: Scalability of N-class attribution in the number of authors (from 2 to 5), using 10-fold cross-validation.

Again, Bayesian networks are consistently superior, while the decision tree shows the poorest performance.

These results are graphed in Figure 3.4, illustrating that the performance of the weaker methods declines sharply. We contend that these results demonstrate that multi-class classification is a much better test of effectiveness than is two-class classification: methods that are more or less indistinguishable for distinguishing between two authors are well separated for the task of identifying one author from amongst many. However, many prior studies have focused on binary AA tasks.

Note, however, that the worst case differs depending on the number of authors. For two-class classification, a random assignment gives 50% accuracy, while for five-class random assignment gives 20%. Thus, while effectiveness does degrade as the number of authors is increased, it is also the case that the problem is becoming innately more difficult.

Side Experiments: The Federalist Papers

As an illustration of the limitations of some previous work on attribution, we experiment with the 65 Federalist papers of known authorship. This corpus has limitations, in addition to the small size, in particular that 50 of the papers are by one author and 15 by another, so that the worst case result—random assignment—is about 64%. However, this is the kind of corpus has been used in much of the previous work in the area.

As shown in Table 3.11, using 10-fold cross validation, results ranged from 76.9% for nearest-neighbor to 95.4% for the decision tree. With this unbalanced experimental setup, apart from the Bayesian networks and the C4.5 decision tree, the other three methods are likely to attribute Madison's works incorrectly. For the naïve Bayesian classifier, with a high overall effectiveness at 89.2%, it fails to attribute Madison's work 53.3% of the time. Whether the differences are statistically significant is unclear. When the problem is further reduced to 15 by each author, all methods but nearest-neighbor (which was inferior) perform excellently, with only one or two errors each. However, while this accuracy is at first sight a success, we believe that it is a consequence of the inadequacy of the test data. Slight differences in assignment lead to large numerical differences in accuracy that are probably not statistically significant. Since the number of samples is very small, it is not illuminating

Table 3.11: The attribution accuracy as well as confusion matrices of using five methods on the 65 Federalist papers. There are 50 documents that are believed to be written by Hamilton (H) and the other 15 by Madison (M).The bottom part of the table, evaluation is based on 15 documents from each author. Numbers that indicate the correct decisions are in bold.

	NaïveBayes		BayesNet		NN		3NN		C4.5	
Authors	H	M	H	M	H	M	H	M	H	M
H	**50**	0	**49**	1	**48**	2	**50**	0	**49**	1
M	7	**8**	3	**12**	13	**2**	12	**3**	2	**13**
Accuracy	89.2		93.8		76.9		81.5		95.4	
Authors	H	M	H	M	H	M	H	M	H	M
H	**13**	2	**15**	0	**12**	3	**15**	0	**15**	0
M	0	**15**	1	**14**	1	**14**	1	**14**	1	**14**
Accuracy	93.3		96.7		86.7		96.7		96.7	

to do a significance test in this case. In contrast, we observed statistical significance for even small numerical differences in the previous experiments, due to the large number of documents involved. Although similar sets of test data have been widely used in previous work, we believe the observed results may not be reliable.

One-class Authorship Attribution

We then examined the effectiveness of each method for one-class classification, using 10-fold cross validation consistently. A reasonably large number of negative samples are required to experiment with one-class AA. The APoc collection is therefore used; anonymous documents in AP are included in this collection. Results, shown in Table 3.12 and Figure 3.5, are averaged across all seven authors. The effectiveness is measured on positive examples. In each block of the table we had a fixed number of positive documents per author and varied the number of negative documents. For small scale, only 25 positive samples are used for each of the seven authors; 400 positive samples are used as for large scale experiments. As discussed previously, this problem is inherently harder than the problems considered above,

Table 3.12: Effectiveness (percentage of test documents correctly attributed) of each method for attribution, using cross-fold validation on one-class classification. Effectiveness is measured on only the positive examples.

Number of negative samples	NaïveBayes	BayesNet	NN	3-NN	C4.5
Given 25 documents per author					
25	93.7	86.9	96.6	97.7	78.9
50	83.4	80.0	94.9	95.4	72.6
100	64.0	73.1	72.0	64.0	65.1
200	47.4	65.7	63.4	54.3	53.7
400	36.0	50.9	58.3	44.0	47.4
600	31.4	46.3	52.6	38.9	34.3
800	29.1	44.6	49.3	37.1	30.3
1200	27.9	41.1	46.3	36.0	29.7
1600	22.3	38.7	41.0	30.1	25.7
Given 300 documents per author					
25	96.7	98.4	99.8	100.0	97.1
50	94.2	96.9	99.6	100.0	94.1
100	87.1	94.0	96.4	98.8	90.4
200	83.9	90.2	92.2	94.7	84.4
400	80.5	86.7	87.3	87.6	78.8
600	78.1	83.2	83.1	83.1	74.7
800	73.9	81.1	82.2	82.6	70.6
1200	72.8	79.3	81.0	79.3	65.7
1600	72.8	78.9	78.5	76.9	61.3

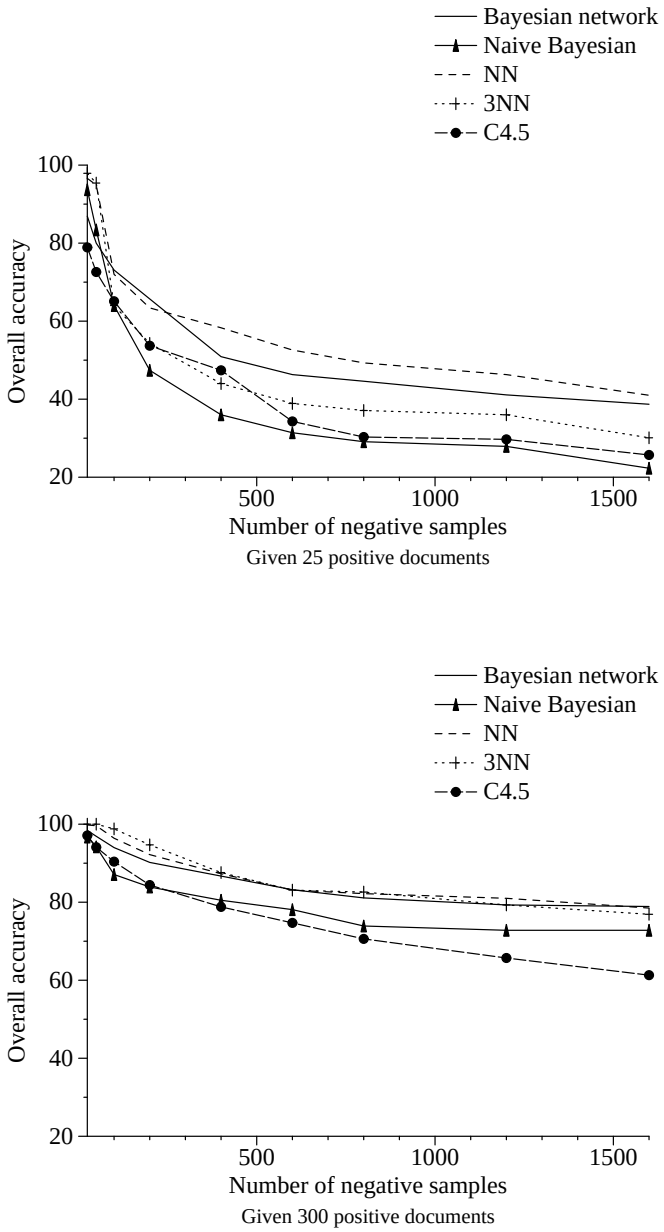


Figure 3.5: Scalability of one-class classification, as the number of negative samples is increased.

as the noise documents are not by a limited set of authors, and thus do not share style.

The results show that the accuracy declines significantly as the number of noise documents is increased. As expected, the effectiveness degrades much faster with a small number of positive samples than with more positive samples. The best methods—nearest-neighbor for a small set of positive examples and Bayesian networks and both nearest-neighbor methods for a larger set of positive examples—are markedly better than the alternatives. In contrast, decision trees are poor for both cases.

This experiment is in our view the most representative of attribution on a large collection, and has moreover shown the most power to distinguish between methods. We contend therefore that one-class classification may be a better test of an attribution method. Again, we use the student t-test to examine whether the effectiveness of difference approaches is statistically different; test results are presented in Table 3.13. As shown, the differences between different pairs of methods are mostly significant.

These experiments have also shown that attribution is indeed reasonably effective. In even the most difficult case, the best methods are shown to be reasonably scalable as the number of documents is increased, with for example an accuracy of around 50% when only 2% of the training documents were positive examples.

Computational Cost

Although efficiency is not the primary concern in this thesis, it becomes important as collection sizes increase. The next experiment is to time some of the methods we used. SVMs used in our experiments are implemented in *C*, while other classifiers are implemented in *Java*. In this respect, we only time the five methods in Weka package, which are directly comparable. In terms of computational complexity, the Bayesian networks are the most costly amongst the five methods; the expected cost increases exponentially as the network structure becomes more complex. We hope to obtain an indication of the cost required for each classification method by timing the experiments. These times are shown in Table 3.14, separated into training time and per-document attribution time. While they cannot be taken as conclusive, they do provide an indication of how well each approach scales. We can observe that the times do not strongly depend on whether the examples are positive or negative. Bayesian net-

Table 3.13: The student t-test between different pairs of methods on the one-class attribution tasks, at a confidence level of 0.05. The "difference" is calculated by subtracting the overall effectiveness of "method 2" from "method 1".

# of samples	method 1	method 2	difference	p-value	significant?
25 positive samples					
	NaïveBayes	BayesNet	-7.4%	0.037	Yes
	NaïveBayes	NN	-14.0%	$6.451e^{-5}$	Yes
	NaïveBayes	3-NN	-7.1%	$4.18e^{-5}$	Yes
	NaïveBayes	C4.5	2.1%	0.459	No
	BayesNet	NN	-6.5%	0.003	Yes
	BayesNet	3-NN	0.3%	0.922	No
	BayesNet	C4.5	9.5%	$1.797e^{-6}$	Yes
	NN	3-NN	6.8%	0.005	Yes
	NN	C4.5	16.1%	$8.656e^{-7}$	Yes
	3-NN	C4.5	9.2%	0.011	Yes
300 positive samples					
	NaïveBayes	BayesNet	-4.8%	$2.373e^{-5}$	Yes
	NaïveBayes	NN	-6.2%	$1.996e^{-6}$	Yes
	NaïveBayes	3-NN	-6.6%	$1.685e^{-5}$	Yes
	NaïveBayes	C4.5	2.1%	0.132	No
	BayesNet	NN	-1.4%	0.001	Yes
	BayesNet	3-NN	-1.7%	0.019	Yes
	BayesNet	C4.5	6.9%	0.002	Yes
	NN	3-NN	-0.3%	0.437	No
	NN	C4.5	8.3%	$1.679e^{-4}$	Yes
	3-NN	C4.5	8.6%	$3.883e^{-5}$	Yes

Table 3.14: Times (milliseconds) for each of the methods. Results in each column are total training time on the left and per-document classification time on the right, in a one-class experiment. Times are averaged over 70 runs.

Examples		Classifier				
Positive	Negative	Naïve Bayes	Bayes net	NN	3-NN	Decision tree
25	25	141/53	4,513/12	20/86	20/100	310/2
25	400	490/38	16,211/8	60/764	50/797	1517/1
300	25	301/28	16,657/7	40/442	30/492	1060/1
300	400	581/25	76,392/8	60/930	60/1,033	3,696/1

works do have by far the greatest training time, and the cost of training grows super-linearly. Training time for the other methods is smaller.

However, the per-document classification times are less consistent. Bayesian networks and decision trees are fast, while for the larger collections the nearest-neighbor methods are over a hundred times slower. Given the relatively poor effectiveness of the naïve Bayesian classifier and the decision tree—the only methods that are fast for both training and classification—choices of method in practice will depend on the applications.

Refinement of one-class Experiments

In all previous experiments, we use the complete set of 363 function words. It is often the case that given a limit number of positive samples available in one-class AA, a large proportion of the function words may not be used by a particular author at all, so that lots of features have zero occurrence, meaning that we know nothing about the habit in use of these rarer words of a particular author. The effect of non-zero features maybe overwhelmed by zero features, and thus may distort the classification. On the other hand, some function words are fairly frequent, even with a small number of documents. An example is shown in Table 3.15; given a 100-document set, there are 57 function words occurring more than five times in at least

Table 3.15: The number of function words that occur more than 5 and 10 times in a documents for each of the 7 authors (A number of 100 documents are randomly selected for each author).

Threshold	Schweid	Currier	Skidmore	Dishneau	Kendall	Crutsinger	Beamish
5	77	69	57	96	82	87	92
10	29	24	26	40	32	35	34

one document for Skidmore, while there are 96 such words for Dishneau. The threshold, 5, indicating how frequent a function word is used. With the APoc collection, we further generate two feature sets that consist of 313 and 176 function words, by setting 5 and 10 as the threshold respectively. We name these two feature sets *subset-313* and *subset-176*.

Authors have different habits in their use of function words, even common ones. In subset-313, each function word is used more than five times in at least one document by any one of the seven authors; amongst all 313 function words, only 45 are shared by all of the 7 authors. In subset-176 on the other hand, only 18 function words are used more than 10 times in at least one documents by all authors. We re-experiment with the one-class AA investigation by applying subset-313 and subset-176 feature set. The purpose of this investigation is to examine whether using only the most frequent function words can improve the AA effectiveness, since as reviewed in Chapter 2 many earlier studies have been restricted to such features [Baayen et al., 2002; Binongo, 2003; Juola and Baayen, 2003]. Results are shown in Figure 3.6 and 3.7 in relation to small numbers of positive samples and large numbers of positive samples. The overall results are averaged from all possible combinations; nonetheless, the influence of removing rare function words is task and method dependent.

Given only 25 positive samples for evaluation, the effectiveness of three methods is more or less improved by using common function words: naïve Bayesian, nearest-neighbor, and 3-nearest-neighbor. In particular, great improvement is achieved for the naïve Bayesian method using the subset-176 feature set. Given 1,600 negative documents, the accuracy is improved by more than 25% compared to the use of the original feature set, that is nearly doubled. However using subset-313 leads to little improvement with the naïve Bayesian method. For the two instance-based learning methods, that is, nearest-neighbor and 3-nearest-neighbor,

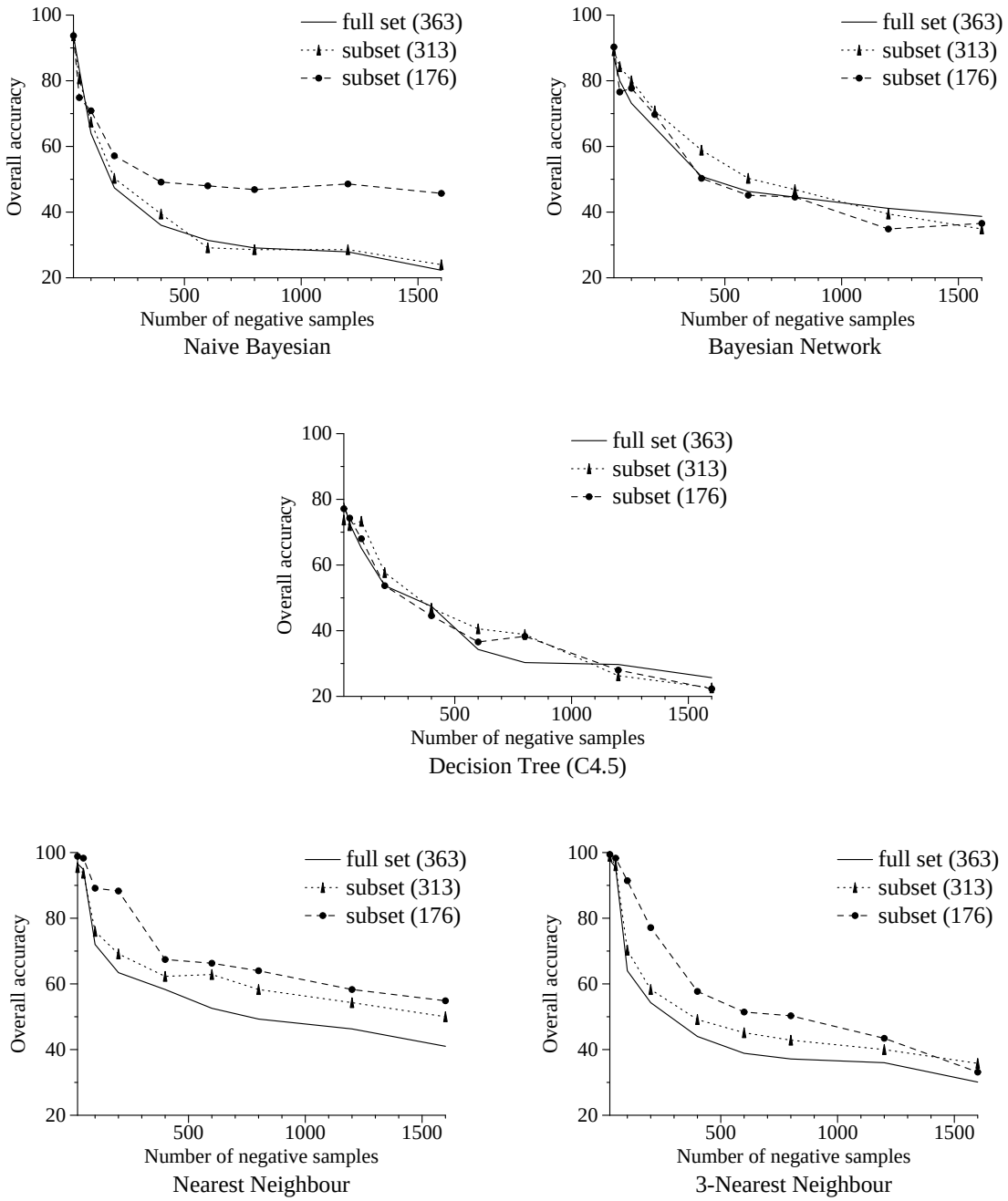


Figure 3.6: The results of using two sets of common function words. All five methods are given 25 positive samples that are consistent with previous one-class AA.

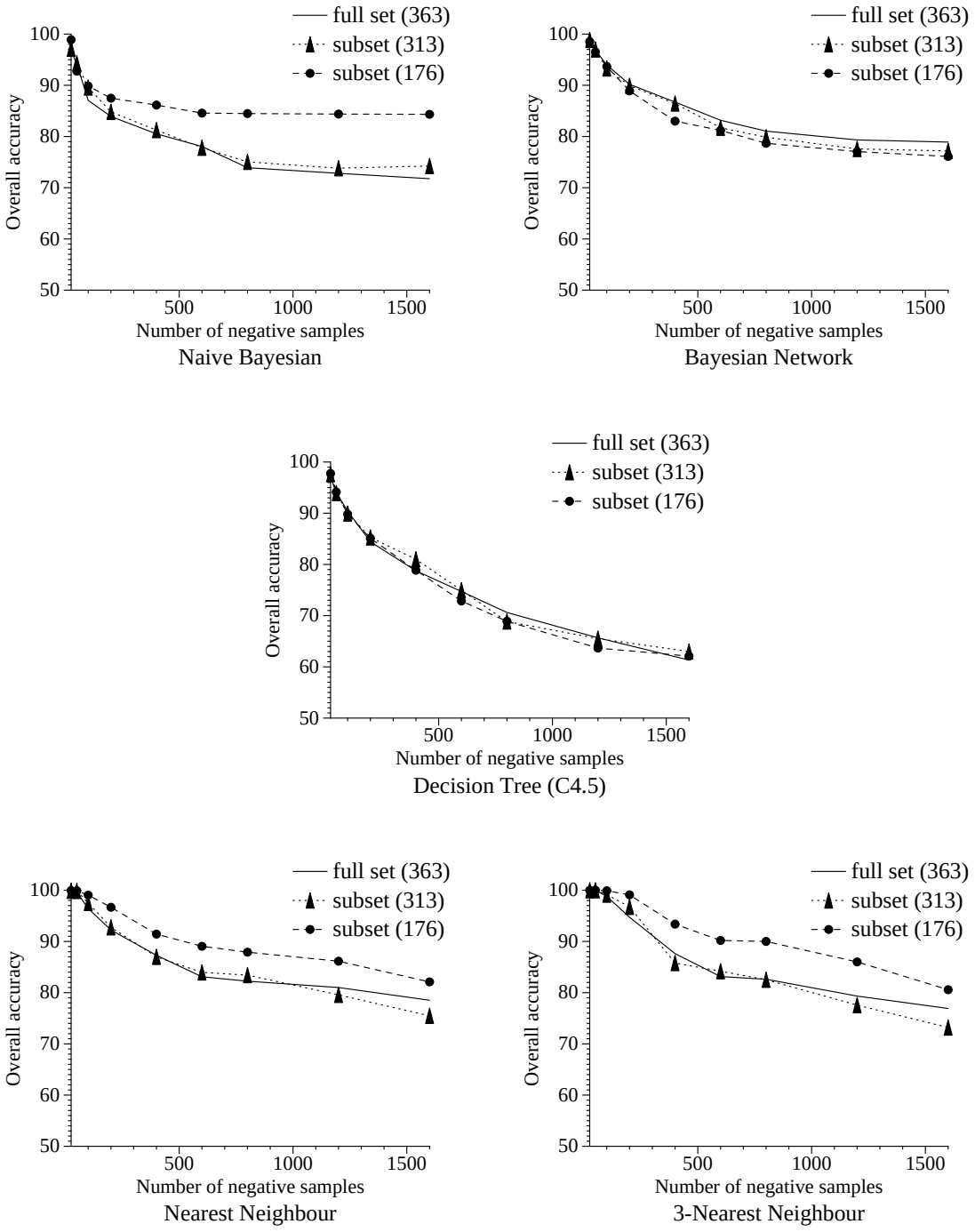


Figure 3.7: The results of using 2 sets of common function words. All five methods are given 300 positive samples that are consistent with previous one-class AA.

both refined feature sets are helpful, while the improvement is not as substantial as achieved by naïve Bayesian. The subset-176 feature set is consistently better than subset-313 for these two methods. Interestingly, with Bayesian networks and decision tree, little benefit is obtained by either of the new feature sets.

On the other hand, we observe different trends in large scale experiments, given 300 positive samples in comparison with 25. All methods are more effective as more positive samples are included. For naïve Bayesian, the changes are consistent with those observed in small scale experiments. There is almost no improvement when applying subset-313 to both nearest-neighbor and 3-nearest-neighbor methods. Moreover, the effectiveness even degrades when the number of negative documents is over 1000. In contrast, subset-176 does improve the effectiveness significantly. Interestingly, Bayesian networks are worse with both refined feature sets. Decision trees are consistently inferior compared to other alternatives; little improvement is observed with common function words.

The results show that feature selection is a task and method dependent process. It is not always ideal to pre-define a fixed selection on features that can satisfy any AA task.

3.3 Chapter Summary

In this chapter we have discussed the development of test collections that can be suitable for evaluating authorship attribution techniques. Several collections are developed and introduced: six collections are developed from The Associated Press (AP), a sub-collection of TREC data consisting of newswire articles; another 2 are created by downloading the English literature from Project Gutenberg. These collections are designed for different types of AA tasks, amongst which AP7 and APoc are used for the preliminary investigation.

We have also undertaken the first comparison of authorship attribution methods on the two collections, using five competitive machine learning methods from the area of TC. These results are the baseline results in this thesis. The results have shown that a consistent test corpus can be used to distinguish between different approaches to attribution. Both AP7 and APoc are suitable for AA and can be used as standard data. However, it is also important to design experiments appropriately. Results need to be averaged across multiple experiments, as some authors are easier to attribute than others. We have also found that one-class

attribution provides the greatest discrimination between methods.

For binary authorship attribution, the SVMs performed the best. For other attribution tasks, the Bayesian networks have been shown to be the most effective methods that we considered, while the C4.5 decision tree is particularly poor in most of the experiments we reported. We have also found that—given an appropriate classification method—function words are a sufficient style marker for distinguishing between authors, although it seems likely that further style markers could improve effectiveness. The best methods can scale to over a thousand documents, but effectiveness does decline significantly, particularly when the number of positive examples is limited.

We have also carried out experiments to illustrate the limitation of many prior AA studies. The results showed that the PCA is not scalable for relatively large collections or for more than two authors. We also experimented with the Federalist paper collection as a second illustration. All five methods are effective; however, we suggest that it is not a plausible corpus, not only due to the limited data, but also the skewness. Even with one or two misclassifications, the reported accuracy can be very different.

There are many alternatives that have been proposed for authorship attribution (AA), including methods based on compression. However the compression-based AA techniques have been controversial, and therefore we do not consider them for a baseline. The effectiveness of such techniques is currently unknown. A novel AA approach based on information theory is proposed in the next chapter, the results from which are compared to that by SVMs and the Bayesian networks, which have been the best methods in our preliminary investigation.

Chapter 4

Relative Entropy for Authorship Attribution

In Chapter 3 we undertook a preliminary investigation of authorship attribution (AA), designed to explore whether the selected 6 machine learning techiques are effective. We evaluated several methods including six machine learning classifiers, on multiple data sets, and established the baseline results to be used in this thesis. As observed, none of the selected techniques were particularly satisfactory in terms of effectiveness, scalability, or efficiency.

The major contribution presented in this chapter is that we propose a principled approach for AA,which is able to accommodate various underlying feature selection and probability estimation methodologies. The approach is motivated from information theory: we explore how Kullback-Leibler divergence—also known as relative entropy—can be used to measure similarities between documents in terms of the writing style; and, the approach outperforms existing techniques in several respects.

Results on *AP7* and *GutenbergSmall* show that, with binary AA, our entropy-based approach significantly outperforms existing machine learning methods, including SVMs, with relatively little training data; however, SVMs are slightly (though not statistically significantly) better when more training samples are included. For multi-class AA, our method is superior to SVMs. In addition to the better effectiveness, our model has lower computational cost and is cheaper to train compared to the selected machine learning techniques. Finally

the results show that such use of entropy is a promising alternative for other categorization problems, and provides an interesting point of comparison: it is directly inspired by information theory, computationally simple, and effective.*

4.1 Background

In the area of AA, a range of classification-based attribution methods have been proposed. However due to the small collections used, the results are not reliable, as discussed in Chapter 3.

Principal component analysis (PCA), a technique based on clustering, has been used in many earlier AA studies to investigate writing patterns in documents [Baayen et al., 2002; Binongo, 2003; Holmes et al., 2001]. However, as shown in Chapter 3, PCA has its limitations in distinguishing between more than two authors and with collections of a large number of documents; a similar observation was reported by Hoover [2001].

Other alternatives to AA are machine learning approaches; they are reasonably effective. However, our preliminary results have shown that these methods also have defects, not only in terms of effectiveness, but also in efficiency and scalability. Consider support vector machines (SVMs), one of the best classification techniques in machine learning. SVMs have been recently applied to AA in several studies [Diederich et al., 2003; Fung, 2003; Koppel and Schler, 2004], and have achieved promising results. However, the computational complexity of SVMs is high; the optimization problem in use of SVMs is quadratic, and the state-of-art learning algorithm for SVMs has a computational cost of $O\left(kn^2\right)$, where n is the number of training samples and k is the size of the feature space. This indicates that SVMs are not always ideal in terms of efficiency. However, to achieve a plausible effectiveness, SVMs usually require reasonably large data—both samples and features—for training; in a typical scenario of AA, it is not always feasible to collect large volumes of materials written by certain authors. In addition, SVMs are not directly suitable for multi-class or n-class classification; as described in Chapter 2, SVMs usually transform the n-class classification problem to a total of n binary cases, which are relatively easier tasks.

N-grams or Markov chains at the character level have been used for identification of

*This chapter incorporates work originally published by Zhao et al. [2006].

writers [Khmelev and Tweedie, 2002; Kešelj et al., 2003]. Such methods construct an $m \times m$ transition matrix from a document or groups of documents for computation, where m is the number of distinct tokens, such as characters or words. One of the primary issues in use of such methods is the computational complexity; it increases exponentially when N increases, and quadratically when m increases, where N is the number of consecutive characters used in N-grams and m is the number of distinct characters. Therefore, it is computationally inefficient to use N-gram Markov chains with large feature sets or long N-grams. Also, when N increases, there will be no occurrences of many features in the document, particularly when the document is not long.

Recently, several investigations have reported use of N-grams for AA. Kešelj et al. [2003] used N-grams to compute similarities between authors' profiles. In their experiment with English data, three authors were selected to be differentiated; each only had one book for training data; a total of 8 books from 8 authors were to be assigned to one of the three potential authors. With bi-grams on 3-class AA—using 676 (26×26) features—the effectiveness was only 67%. There are also investigations using collections in languages other than English, such as Greek [Stamatatos et al., 2000], Chinese [Peng et al., 2003b], and Russian [Kukushkina et al., 2001].

N-grams have been used in compression algorithms for AA [Benedetto et al., 2002]; however, such approaches have been controversial. As summarised in Table 2.4, the effectiveness of compression for AA is not stable; it varies significantly, and is subject to the compression programs used. Compression algorithms build a model of the data; a coding technique then uses the model to produce a representation of the data. Typically, the compression methods are designed for reasonable speed and introduce many approximations into their models, and thus may not provide a good indication of the characters of the underlying model. Also, compression is based on modelling of character sequences, so there is a bias introduced by the subject of the text; the model is therefore highly approximate surrogate for an underlying model of a probability distribution. In this respect, much accuracy may be lost, and therefore little can be learned about which aspects of the modelling lead to the success of AA. Goodman [2002] and Khmelev and Teahan [2003a] have criticized the work reported by Benedetto et al. [2002] based on compression in different perspectives, as

discussed in Chapter 2. Therefore, we do not use compression-based AA in this thesis.

In this chapter, the main contribution of our research is to propose a novel AA method that outperforms existing techniques. In the following sections we describe two key components of our method: information theory, in particular the Kullback-leibler divergence, and language models, together with smoothing techniques.

4.2 Information Theory and Entropy

Information theory was originally introduced in the study of electrical communication [Shannon, 1948], and subsequently applied to a variety of other fields. Information theory has been used to explore properties of English texts. In the context of English, terms can be written in the some relative frequencies and of the probability that one word follows another word, word pairs, or other word combinations. The statistical properties of sequences of characters, or words, of English text can be generated by a sequence of random choices among these terms or letters. The choice of terms and letters depends on the probabilities of their occurrence in the sequences.

Suppose we have a set of possible events of a random variable X, $\{x_i \in X | i = 1, 2, \ldots, n\}$, whose probabilities of occurrence are $p(x_i)$. The concept of entropy in information theory was introduced to measure how much choice is involved in selection of the events, and how uncertain the outcome is in relation to the random variable X.

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{4.1}$$

$$\text{where} \quad 1 = \sum_{i=1}^{n} p(x_i) \tag{4.2}$$

Equation 4.1 presents the standard formula of measuring entropy, where $H(X)$ is the entropy of random variable X and $p(x_i)$ is the probability of the event x_i, that is, a character or a word in this case. The value of $p(x_i)$ is usually derived from a probability mass function that also satisfies the constraint of Equation 4.2. The value of $H(X)$ is the average number of bits that is required to represent each symbol or event x_i in the random variable X. In theory, the better the model, the smaller the number of bits needed. The entropy is themaximum when each x_i occurs with the same probability.

We use an example to illustrate use of this principle in the context of English text. An entropy model can be built for a collection of documents by identifying the following factors:

- $W = \{w_i | i = 1, \ldots, n\}$: a set of distinct words over an entire data collection. W is a random variable, and each word w_i is one of the events within the random variable space.
- $F = \{f(w_i) | i = 1, \ldots, n\}$: a set of frequencies. Each $f(w_i)$ is the number of times that the word w_i occurs in the collection.
- $N = \sum_{i=1}^{n} f(w_i)$: the total number of word occurrences in the collection.

To build a context-free model, where no use is made of word order, the probability of each $f(w_i)$ can be approximated in a straightforward manner, that is, the maximum likelihood: $p(w_i) = f(w_i)/N$. The entropy of W is then computed as:

$$H(W) = -\sum_{i=1}^{m} \frac{f(w_i)}{N} \log_2 \frac{f(w_i)}{N} \tag{4.3}$$

where $H(W)$ indicates an average number of binary digits required for representing each w_i. Therefore by using this model, the minimum number of bits required to represent the entire collection is $N \times H(W)$.

Entropy provides a flexible way of modelling: a model can be built for each document individually in the collection, or for any number of documents as a whole, given the corresponding probability mass functions. However a difficulty in using direct entropy measurements on a new document is that it may contain a word w' that is absent from the original model, leading to $p(w') = 0$ and undefined $\log_2 p(w')$. We examine this issue in Section 4.4.

4.3 Relative Entropy: Kullback-Leibler Divergence

Another way to use entropy is to compare two models, that is, to measure the difference between two random variables. In contrast to entropy, which is concerned with the sample distribution of a single random variable, relative entropy is concerned with the relation between sample distributions of two random variables, and measures how different these two

variables are. A mechanism for measurement of relative entropy is known as the *Kullback-Leibler divergence* (KLD) [Kullback and Leibler, 1951; Manning and Schütze, 1999].

Suppose two random variables X_p and X_q have probability mass functions θ_p and θ_q. Both variables have the same set of events $\{x_i | i = 1, \ldots, n\}$; θ_p generates a set of probabilities $\{p(x_i) | i = 1, \ldots, n\}$, in which each $p(x_i)$ is the probability of the event x_i occurring in X_p; and similarly, θ_q generates probabilities $\{q(x_i) | i = 1, 2, \ldots, n\}$. The relative entropy or Kullback-Leibler divergence is defined to be:

$$KLD(\theta_p || \theta_q) = \sum_{i=1}^{n} p(x_i) \log_2 \frac{p(x_i)}{q(x_i)} \tag{4.4}$$

The quantity of the divergence (KLD) between θ_p and θ_q is non-negative, which can be interpreted as the average number of bits that is wasted by converting from the distribution of θ_p to the distribution of θ_q. For this reason, KLD can be considered as a distance between the two probability mass functions, as KLD provides a measure of how close these two probability mass functions are.

In this chapter, we propose the use of KLD as a categorization technique. The principle of the KLD-based approach is straightforward. Either individual documents can be regarded as random variables, or a group of documents can form a single random variable; features extracted from a document or a group of documents are the events of a random variable of that document. If the probability mass function $\theta_{d'}$ that measures probabilities of the event occurrence of a document d' is closer to the probability function θ_p of a document or a group of documents d_p, than to θ_q of d_q, that is, the divergence between the document d' and d_p is smaller than that between document d' and d_q:

$$KLD(\theta_{d'} || \theta_p) < KLD(\theta_{d'} || \theta_q)$$

then we hypothesize that the document d' is more similar to d_p than d_q. If d_p and d_q indicate different classes, then d' is assigned to the same class as d_p rather than d_q in this case. Thus, to use relative entropy for classification, it is necessary to find a probability mass function that measures the probability distribution of features occurring in a document or a group of documents. In the next section, we introduce how to use language models to approximate the probability distribution.

4.4 Language Models and Smoothing

Language models have been extensively used in speech recognition [Mori and Brugnara, 1996] and widely used in information retrieval in recent years. The purpose of language models is to estimate the distribution of terms in natural language units. In information retrieval (IR), language models perform at least as well as, if not better than, alternatives based on vector space or probabilistic models [Zhai and Lafferty, 2004].

In the context of IR, and given a document d, a language model $\hat{\theta}_d$ built from the document d can be used to measure the probability that $\hat{\theta}_d$ could have generated an input query q [Croft and Lafferty, 2003]. However, as a single document does not usually contain all the vocabulary of a collection, and in particular may not contain all of the query terms, it is problematic to use the model $\hat{\theta}_d$ to estimate the probability of these missing terms occurring in d. To address this issue, smoothing techniques have been proposed [Chen and Goodman, 1996; Hiemstra, 2002; Zhai and Lafferty, 2001a; 2004].

While a great diversity of smoothing methods have been proposed, the principle idea of any smoothing process is to assign non-zero weights to zero-occurrence features. All smoothing methods work by reallocating some of the probability mass of features that appear in a document to the unseen features. Some smoothing techniques simply use additional counts to represent unseen terms, while others may make use of the entire document collection as a background model to provide an estimate. Researchers have investigated smoothing in relation to standard ad hoc information retrieval tasks [Zhai and Lafferty, 2001a; 2004]. In the context of ad hoc retrieval, a common assumption is that the characteristics of the rest of the documents in the collection are in some sense similar, and can be collectively used to estimate unseen features of the document in question.

In the following sections we mathematically describe language models and several popularly used smoothing techniques. These are two key components of our classification model, which estimates probabilities of feature occurrence. In order to derive a precise methodology, some atomic components are defined as follows.

Atomic Components. A testbed consists of texts. In the AA context, these texts are from certain authors. Style markers are the feature units extracted to represent the writing

style of these documents. The probability function of style markers is derived by combining simple statistics or primitive information from the collection, such as:

- A set of features or style markers $F = \{f_i | i = 1, \ldots, k\}$; each document is then a sequence of features.
- A document d; $|d|$ is the number of features that can be extracted from the document d.
- A particular author a.
- A group of documents d^a written by the author a; $|d^a|$ is the total number of features that can be extracted from this training set.
- The number of times $f_{i,d}$ that the ith feature occurs in the document d.
- The notation $\hat{\theta}$ refers to an estimated model.
- A language model $\hat{\theta}_d$ built from document d.
- A language model $\hat{\theta}_a$ built for author a.
- A document collection C; $|C|$ is the total number of features extracted from C.
- A background language model $\hat{\theta}_B$ built for smoothing.
- Smoothing parameters δ, λ, and μ (details are presented in Sections 4.4.1 to 4.4.4).

A straightforward estimation in language modelling is the maximum likelihood estimate, in which the probability of each feature is given by its frequency, normalized by the total number of features in that document (or, equivalently, the category). Here, categories are labelled by the potential authors; each author represents one category. The probability of each feature f_i in document d can be estimated by the document model $\hat{\theta}_d$ as:

$$p_{\hat{\theta}_d}(f_i) = \frac{f_{i,d}}{|d|} \tag{4.5}$$

The probability of a feature f_i occurring in the training document d^a of a particular author a can be measured in a similar manner:

$$p_{\hat{\theta}_a}(f_i) = \frac{f_{i,d}}{|d^a|} \tag{4.6}$$

Thus the divergence (KLD) between the document model $\hat{\theta}_d$ and an author model $\hat{\theta}_a$ can be measured by:

$$\begin{aligned} KLD\left(\hat{\theta}_d||\hat{\theta}_a\right) &= \sum_{f_i \in F} p_{\hat{\theta}_d}(f_i) \log_2 \frac{p_{\hat{\theta}_d}(f_i)}{p_{\hat{\theta}_a}(f_i)} \\ &= \sum_{f_i \in F} \frac{f_{i,d}}{|d|} \log_2 \frac{f_{i,d} \cdot |d^a|}{f_{i,d} \cdot |d|} \end{aligned} \tag{4.7}$$

However, it is usually the case that some features or style markers are unseen in either the training documents—that is, a group of documents available by a particular author a—or the documents to be attributed or classified. This introduces an undefined value caused by zero in the denominator in Equation 4.7, and thus, the value of KLD cannot be calculated. This is a standard problem with such models. Researchers have explored a variety of smoothing techniques [Zhai and Lafferty, 2004] to calculate the probability of unseen f_i. The principle of any smoothing process is to assign non-zero values to zero occurrence features. All smoothing methods work by reallocating some of the probability mass of features appearing in a document to the unseen features.

In the ad hoc retrieval context, a common assumption is that the characteristics of the rest of the documents in the collection are in some sense similar, and can collectively be used to estimate unseen terms in the document in question. In theory, a background model could be any source of typical statistics for features or style markers. However, intuitively it makes sense to derive the background model from other documents of the same domains. For instance, in attributing newswire articles, a background model derived from scientific fiction stories seems unlikely to be appropriate. As the background model, we use the aggregate of all known documents within the same domain—that is, the AP newswire, including training, testing, and other unused documents, as this gives the largest available sample of materials.

We briefly review some smoothing approaches below; each uses the statistics collected from a background model for the estimation of the smoothed probability of each feature f_i in the document d. In this research, we explore four techniques that are reported to be effective in the area of IR. Note that we do not claim to thoroughly investigate the smoothing approaches.

4.4.1 Absolute Discounting

In absolute discounting, a constant δ is deducted from the frequency of each feature that is extracted from the documents [Ney, 1994] . The model is derived as follows:

$$\hat{p}_{\delta,\hat{\theta}_d}(f_i) = \frac{\max(f_{i,d} - \delta, 0) + \delta|d|_u \; p_{\hat{\theta}_B}(f_i)}{|d|} \tag{4.8}$$

where the notation $\hat{p}$ refers to the smoothed probability. The value of δ is between 0 and 1 inclusive, and $|d|_u$ indicates the number of distinct feature components extracted from the document d.

4.4.2 Jelinek-Mercer Smoothing

Jelinek-Mercer smoothing, also known as linear interpolation smoothing [Jelinek and Mercer, 1980], estimates the likelihood of each feature occurrences by referring to both the document model $\hat{\theta}_d$ and the collection model $\hat{\theta}_B$. It uses a coefficient value λ to adjust the input weight of each component f_i from the two different types of models:

$$\begin{aligned} \hat{p}_{\lambda,\hat{\theta}_d}(f_i) &= (1-\lambda) \; p_{\hat{\theta}_d}(f_i) + \lambda \, p_{\hat{\theta}_B}(f_i) \\ &= (1-\lambda) \; \frac{f_{i,d}}{|d|} + \lambda \, p_{\hat{\theta}_B}(f_i) \end{aligned} \tag{4.9}$$

It can be seen that, when λ decreases, then the document model contribution to the estimated probability increases, and vice versa.

4.4.3 Dirichlet Prior Smoothing

Dirichlet priors, also known as Bayesian Smoothing [Mackay and Peto, 1995], assume counts $\mu \times p_{\hat{\theta}_B}(f_i)$ from the collection model for each feature, and is calculated by:

$$\hat{p}_{\mu,\hat{\theta}_d}(f_i) = \frac{f_{i,d}}{\mu + |d|} + \frac{\mu}{\mu + |d|} \; p_{\hat{\theta}_B}(f_i) \tag{4.10}$$

As seen from the formula:

$$\lim_{|d|\to 0} \frac{\mu}{\mu + |d|} = 1$$

where given fixed μ, for short documents the background probabilities dominate, on the principle that the evidence for the in-document probabilities is weak. As document length grows,

the influence of the background model diminishes. Like the other smoothing parameters, the choice of an appropriate value for μ is a tuning stage in the use of this model.

4.4.4 Two-Stage Smoothing

The two-stage smoothing approach has recently been proposed and applied in information retrieval tasks by Zhai and Lafferty [2004]. It has shown to be better than individual smoothing methods in some cases. In the first stage, a document language model is first smoothed using a Dirichlet prior, and in the second stage, further smoothed by Jelinek-Mercer smoothing.

The two-stage method is motivated by the observation that the Dirichlet smoothing and Jelinek smoothing work well in modelling documents and queries respectively. Therefore we also explored the application of this method to authorship attribution. The model is a combination of Equation 4.9 and Equation 4.10:

$$\hat{p}_{\mu,\lambda,\hat{\theta}_d}(f_i) = (1-\lambda)\left(\frac{f_{i,d}}{\mu+|d|} + \frac{\mu}{\mu+|d|}p_{\hat{\theta}_B}(f_i)\right) + \lambda p_{\hat{\theta}_B}(f_i) \tag{4.11}$$

4.5 KLD as a Classifier for Authorship Attribution

To attribute authorship to a document, training samples are required for each of the potential authors. Straightforwardly, the KLD-based approach builds a model for each author by aggregating the training documents; this kind of model is referred to as the author model. Divergence is then calculated between the document model and each of the author models individually. The author whose model contributes the smallest divergence is identified as the origin of the document. The final classifier for AA is derived by incorporating the probability estimation function into the Equation 4.7. We formulate the classifier as follows, using Dirichlet smoothing for illustration:

$$A = argmin_{\,a}\left(KLD\left(\hat{\theta}_d||\hat{\theta}_a\right)\right) \tag{4.12}$$

$$\text{where } KLD\left(\hat{\theta}_d||\hat{\theta}_a\right) = \sum_{f_i\in F}\left[\left(\frac{f_{i,d}}{\lambda+|d|} + \frac{\lambda}{\lambda+|d|}p_{\hat{\theta}_B}(f_i)\right) \times \log_2 \frac{\frac{f_{i,d}}{\lambda+|d|} + \frac{\lambda}{\lambda+|d|}p_{\hat{\theta}_B}(f_i)}{\frac{f_{i,d^a}}{\lambda+|d^a|} + \frac{\lambda}{\lambda+|d^a|}p_{\hat{\theta}_B}(f_i)}\right]$$

In order to avoid zeroes in the computation, both training and testing documents also contribute to the background model $\hat{\theta}_B$. Although this approach is straightforward, and based

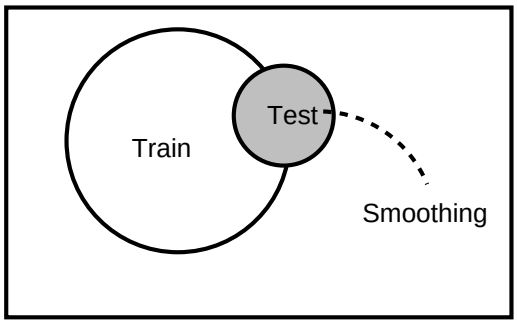


Figure 4.1: Illustration of smoothing applied in IR; gray area represents the terms or tokens involved in smoothing. The rectangle indicates the distinct tokens extracted from the entire collection; the white circle indicates the tokens occurring in the training data.

directly on fundamental principles, to our knowledge it has not been applied to classification problems, including AA.

In IR, smoothing is usually query-centric, that is, only query terms are involved in the smoothing process; terms present in a document but missing in a query are discarded. From a classification viewpoint, a document to be classified is considered as a query in IR; and the training set is individual documents in the collection in IR. Therefore, an example of applying a query-centric smoothing is as in IR is depicted in Figure 4.1.

An information retrieval system returns a ranked list of relevant documents in response to a query provided by a user. However the queries that users enter cannot be predicted beforehand, so that it is not feasible to pre-define any components other than those in the query. In this sense, it is reasonable to just smooth query terms in an IR system.

However, in AA, documents are represented by sequences of style markers—that is, a fixed set of features for all authors, which are usually pre-defined for supervised classification. A certain number of documents of a particular author are used for training, and then used to attribute the test documents whose authorship is to be identified. Similarly, these individual test documents are regarded as query documents. It is commonly the case that some pre-defined style markers are missing in both query documents and training documents.

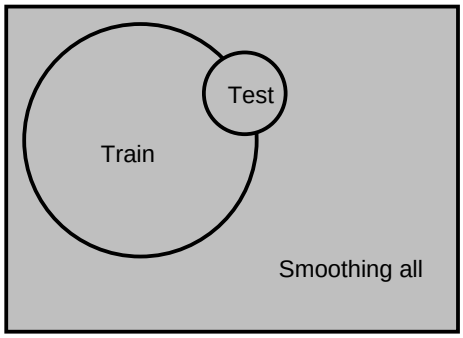


Figure 4.2: The illustration of smoothing that is applied to the entire set of pre-defined style markers

Recall from what we have observed from Chapter 3 that using frequent function words does not guarantee a better result. Therefore, it is worth exploring whether using query-centric smoothing is sufficient for AA, and whether the style markers absent in test documents are actually helpful indicators.

In theory, any feature f_i included in the pre-defined feature set F should be considered. However when a feature or style marker f_i is missing in both the query document d and the training documents a_k, its probability is merely determined by the background model in some smoothing techniques, such as Jelinek-Mercer smoothing. For such smoothing, as shown in Equation 4.12, the probability of f_i estimated by the query document model $\hat{\theta}_d(f_i)$ is then identical to that estimated by the author model $\hat{\theta}_{a_k}(f_i)$, having no effect on quantifying the KLD. In this case, for such method, instead of using all features $f_i \in F$ to measure the KLD, only features that satisfy $f_i \in (d \cup d^a)$ are computed, for the concern of computational cost. Figure 4.2 shows smoothing applied to the entire pre-defined set of style markers.

4.6 Experiments

We conduct experiments on a range of data sources to examine effectiveness and scalability of the proposed KLD method for authorship attribution. Data collections used in this research are the *AP7* newswire articles and the *GutenbergSmall* corpus of English literature.

Both binary AA and multi-class AA are evaluated, and the effectiveness is averaged from experimenting with all possible author combinations for each type of AA. That is, there are 21 (C_7^2) pairs of authors differentiated in binary AA; 35 author combinations in 3-class and 4-class AA; and again 21 author combinations in 5-class AA.

The proposed KLD-based AA method requires a background model to estimate probability distribution of the feature occurrences. For *AP7* data, the background model is derived from the entire AP collection, consisting of over 250,000 newswire articles. We also use this background model for the *GutenbergSmall* collection. It is likely that deriving a background model from large collections of literature would be a better choice for the *GutenbergSmall* collection since the documents are of similar type. However, the literature we have collected is far from sufficient for a background model, and the AP background model is the best option that we have.

4.6.1 Binary Authorship Attribution

In the first experiment, we explore whether the IR-style smoothing paradigm is also plausible for AA, and compare the four smoothing methods for binary AA from different perspectives, including the smoothing paradigms, and the effectiveness of different smoothing methods for AA.

The comparison of AA effectiveness by smoothing different sets of features is on the *AP7* collection. We randomly select 100 documents per author as test documents, which are separate from the training samples; these documents are the same as those used in Chapter 3. We vary the size of the training document pool from 25 to 600 for each of the seven authors. Therefore, there are in total 21 test sets for binary classification, each consisting of 200 documents from a pair of authors. Function words are consistently used as the style markers to tokenise the documents. We evaluate all four types of smoothing methods on binary AA, using the two alternatives of IR-style smoothing and AA-style smoothing to examine which approach works better for AA.

We first apply all four smoothing methods as in IR. As observed, some author pairs are harder to differentiate in comparison with the others; five out of the twenty-one pairs generally produce much lower effectiveness in our experiments. We refer to these as *difficult pairs*, and

the others as *easy pairs*. To examine the difference between two smoothing paradigms, the four smoothing methods are then applied to the complete list of function words. The difficult pairs and easy pairs are evaluated separately; results presented in the following are the best achievable effectiveness on the basis of carefully tuned parameters.

It is not surprising that, for the easier pairs, little difference is observed using the two different ways to apply smoothing. However, a dramatic difference is obtained with the five difficult pairs; results are presented in Figures 4.3 (smoothing all style markers in the feature set) and 4.4 (smoothing style markers only present in the test documents). As shown, the improvement can be achieved by smoothing more than in-document style markers, particularly with Jelinek-Mercer and two-stage methods, whereas the other two are relatively more stable. Due to the small number of difficult pairs, the results may not be conclusive; however, they indicated that rare style markers could also be informative. Therefore in the following experiments, we consistently apply smoothing to the complete set of function words.

Smoothing Effectiveness

In these experiments, we compare the four smoothing techniques on two data sets: the *AP7* corpus and the *GutenbergSmall* corpus. We increase the number of documents used for training, and tune the parameters carefully to achieve the best results for each smoothing method. As discussed below, this may not always be a sensible thing to do, as it can lead to over-tuning of parameters for a particular collection; however, we are exploring whether any one method is clearly superior to the others. The best overall accuracy achieved for each method is shown in Table 4.1 for AP data, and Table 4.2 for Gutenberg data.

It can be seen that all smoothing methods are effective for AA with appropriate parameters, and very small numerical differences are observed. Additionally, we observed a tendency from this series of experiments—that is, the smoothing parameters are adjusted to bring a stronger smoothing effect from the collection model when the training data is relatively small; the impact from the background model decreases as the volume of training samples increases.

In our experiments, the parameters are tuned within the ranges that have been suggested by Zhai and Lafferty [2004] for IR tasks. It is true that tuning parameters in such an empirical

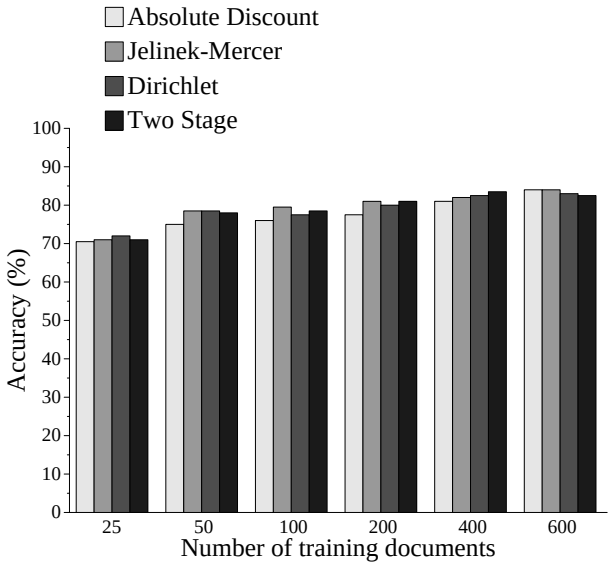


Figure 4.3: The effectiveness of applying smoothing to entire featture set on difficult author pairs with the AP7 collection.

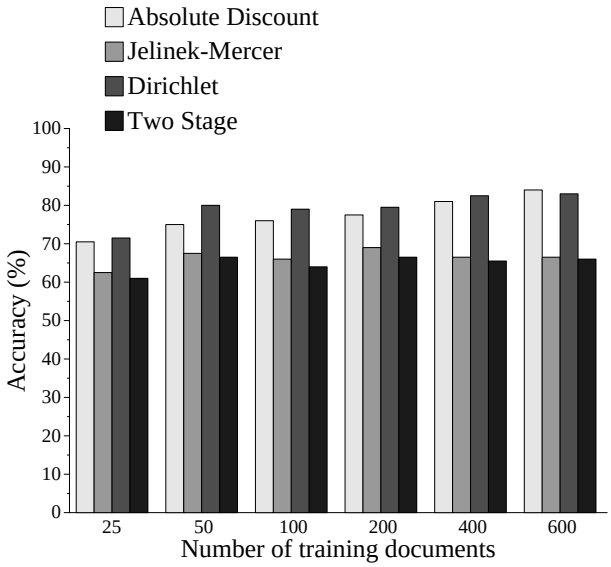


Figure 4.4: The effectiveness of applying IR smoothing on difficult author pairs with the AP7 collection.

Table 4.1: Effectiveness (percentage of test documents correctly attributed) by each smoothing method for two-class classification using the **AP7** *collection. For absolute discounting smoothing, we experiment with* $\delta = [0, 1]$*;* $\mu = [0.01, 10^4]$ *for Dirichlet smoothing;* $\lambda = [0, 1]$ *for Jelinek-Mercer smoothing; and combination of choices of* μ *and* λ *for the two-stage approach.*

Smoothing		# of Training					
collection	methods	25	50	100	200	400	600
AP	Absolute Discount	88.2	89.1	91.5	92.0	92.5	92.7
	Dirichlet	88.2	89.5	91.7	92.3	92.8	92.7
	Jelinek_Mercer	89.1	90.3	92.3	92.7	92.7	92.8
	Two Stage	89.0	89.0	91.0	91.1	91.2	91.0

Table 4.2: Effectiveness (percentage of test documents correctly attributed) of smoothing methods for two-class classification on the **GutenbergSmall** *collection. For Absolute discount smoothing, we experiment with* $\delta = [0, 1]$*;* $\mu = [0.01, 10^4]$ *for Dirichlet smoothing;* $\lambda = [0, 1]$ *for Jelinek-Mercer approach; and combination of choices of* μ *and* λ *for Two stage approach.*

Smoothing		# of Training				
collection	method	25	50	100	200	300
Gutenberg	Absolute Discount	91.7	94.0	94.8	95.4	96.0
	Dirichlet	91.7	94.2	95.7	95.9	96.4
	Jelinek_Mercer	91.8	94.4	95.7	95.9	96.7
	Two Stage	91.9	94.5	95.8	95.9	96.7

Table 4.3: Comparison of smoothing methods on the **AP7** *collection. The p-values for a paired t-test are shown, with the confidence level of 0.05 assigned. The top results are produced when there are 25 documents used for training per author; the bottom results are with more training data, 200 documents per author. In each pair of methods that are compared, the better one is presented in* **bold**.

# of training	M1	M2	pvalue	Significant?
25 (small)	Dirichlet	**Jelinek Mercer**	0.009	Yes
	Dirichlet	Absolute Discount	0.031	Yes
	Dirichlet	**Two Stage**	0.003	Yes
	Jelinek Mercer	Absolute Discount	0.006	Yes
	Jelinek Mercer	**Two Stage**	0.015	Yes
	Absolute Discount	**Two Stage**	0.003	Yes
200 (large)	Dirichlet	Jelinek Mercer	0.197	No
	Dirichlet	Absolute Discount	0.356	No
	Dirichlet	Two Stage	1.000	No
	Jelinek Mercer	Absolute Discount	0.111	No
	Jelinek Mercer	Two Stage	0.031	No
	Absolute Discount	Two Stage	0.370	No

way is not always an ideal way to proceed, since it may cause the parameters to be over-tuned, and thus results in misleading results. Unfortunately, we have not been able to establish a theoretic way to approximate the parameters; this problem is worth investigating in future work. In this work, we do not claim a thorough investigation on smoothing techniques, but intend to explore changes in effectiveness of AA in relation to different underlying probability estimations.

We use a significance test to see whether the small numerical differences between different approaches are in fact statistically significant. The paired student t-test is applied to each possible method pair with different numbers of training samples; we use respectively 25 and 200 training samples for demonstration. Table 4.3 presents the results for the *AP7* collection. We consider values with $p < 0.05$ to be significant.

Table 4.4: Effectiveness (percentage of test documents correctly attributed) for Bayesian networks, SVMs, and KLD attribution on two-class classification. The data is the AP collection, with function words as features. Best results in each case are shown in **bold**.

Docs per author	Bayes network	KLD $\mu = 10^0$	KLD $\mu = 10^1$	KLD $\mu = 10^2$	KLD $\mu = 10^3$	SVM
50	82.0	88.0	**89.7**	85.7	55.6	85.8
100	85.7	91.2	**91.8**	86.3	56.9	89.4
200	88.2	**92.6**	92.4	85.7	54.6	91.1
400	90.6	92.4	**92.8**	85.5	54.6	92.4
600	90.6	92.2	92.7	86.1	55.0	**92.9**

The results of significance tests show that the choice of smoothing methods depends on the number of training documents. With this *AP7* collection, the four different smoothing methods do perform significantly differently when only a small number of training samples is available. Although the overall effectiveness of various methods are fairly close, as shown in Table 4.1, the small numerical difference is significantly different. Jelinek-Mercer smoothing performs better than the other methods. However there is no significant difference amongst these methods when sufficient training data is provided. We then undertake the same test on the *GutenbergSmall* collection. The difference in terms of effectiveness is tiny, and we do not observe any significant difference. The results suggest that, for the best achievable results provided with a sufficient number of training samples, all smoothing approaches can be effective for the estimation of probability distributions of style markers. There is no one smoothing method that significantly outperforms the others.

KLD versus Other Methods

In this experiment, we compare the proposed KLD approach with the baseline methods that have been investigated in Chapter 3, where the Bayesian network and SVMs showed the best performance. We select Dirichlet smoothing in this experiment, as it is effective and reliable, even with difficult pairs. This comparison is carried out with two-class AA on both collections; results are shown in Table 4.4 are on the *AP7* collection, where outcomes are

Table 4.5: The significance test between the KLD-based methods and the best baseline methods, SVMs, on binary AA. The confidence level is set to 0.05. "KLD-SVMs" refers to the numerical differences of between the two methods. The corpus used is AP7.

	50	100	200	400	600
KLD-SVMs (%)	3.9	2.4	1.3	0.4	-0.2
p-value	< 0.001	0.002	0.047	> 0.05	> 0.05
Significant?	Yes	Yes	Yes	No	No

averaged across all 21 pairs of authors. It is observed that the best results are generally obtained for $\mu = 10$.

To examine the scalability of KLD attribution, we increase the number of training documents, and maintain the same set of test documents. As shown, the accuracy of classification increases as the number of documents for training is increased, but appears to plateau. The KLD method is markedly more effective than the Bayesian network classifier. With a small number of documents for modelling, the KLD method is more effective than SVMs, while with a larger number of documents SVMs are slightly superior.

To examine whether the numerical differences are statistically significant, we undertook a series of significance test on an author-by-author basis, with different volumes of training data. Table 4.5 presents the test results on the *AP7* corpus. As shown, our KLD-based method performs statistically better than SVMs with relatively small training samples, up to 200, as observed in our experiments. In addition, in conjunction with the results presented in the Table 3.9, our method is shown consistently better than Bayesian networks with varied sizes of training sets; the differences are statistically significant.

As noted earlier, the computational cost of the SVMs and Bayesian network methods is quadratic or exponential, whereas the KLD method is approximately linear in the number of distinct features. It is thus expected to be much more efficient; however, the diversity of the implementations we used made it difficult to meaningfully compare efficiency.

We also test KLD attribution on the Gutenberg data we had gathered. Average effectiveness is reported in Table 4.6. The trends are similar to those observed for the *AP7* collection. Again, our proposed KLD method is consistently more effective than Bayesian networks, and

Table 4.6: Effectiveness (percentage of test documents correctly attributed) for Bayesian networks, SVMs, and KLD attribution on two-class classification. The data is the Gutenberg collection, with function words as features. Best results in each case are shown in **bold**.

Docs per author	Bayes network	KLD $\mu = 10^0$	KLD $\mu = 10^1$	KLD $\mu = 10^2$	KLD $\mu = 10^3$	SVM
50	93.5	94.0	**94.2**	94.0	74.4	91.4
100	95.1	95.3	**95.7**	95.4	74.0	94.8
200	95.1	95.5	95.9	95.4	74.0	**96.5**
300	95.4	96.0	96.4	96.0	74.8	**97.2**

SVM is more effective than KLD only when a larger number of training documents is used; when SVM is superior, the difference is slight. However we do not carry out the significance test for the *GutenbergSmall* data, since the author-by-author based evaluation only produces 10 results for sampling; applying a significance test is not illuminating in this case. Therefore, in combination these results show that KLD attribution can be successfully used for binary attribution.

4.6.2 Multi-class Authorship Attribution

We next examine the performance of the KLD method when applied to multi-class classification. We compare Bayesian networks and the KLD classification method using function words as the style markers; SVMs are not used, as they cannot be directly applied to multi-class classification.

We use 50 and 400 documents from each author for training with the *AP7* collection, and use 50 and 300 training documents per author with *GutenbergSmall* data. The outcomes are again averaged from all possible author combinations; taking *AP7* for example, the results are averaged from a total of 21 combinations for two and five authors, and 35 combinations for three and four authors. As shown in Table 4.7, with appropriate μ values, the KLD approach consistently and substantially outperforms Bayesian networks. The differences increase when the number of n increases, where n is the number of potential authors; the bigger n, the harder the attribution task is. For a smaller training set—that is, using

Table 4.7: Effectiveness (percentage of test documents correctly attributed) of Bayesian networks and KLD attribution for both AP and Gutenberg data, on two- to five-class classification.

			Number of Authors			
Collection	Training Size	Method	2	3	4	5
AP7	50 per author	BayesNet	86.0	79.5	75.8	71.7
		KLD	89.7	83.9	79.9	76.2
	400 per author	BayesNet	90.1	85.2	80.6	76.3
		KLD	92.2	88.3	84.9	82.2
GutenbergSmall	50 per author	BayesNet	93.5	88.8	87.7	86.0
		KLD	94.2	91.6	89.1	87.0
	300 per author	BayesNet	95.4	91.1	88.8	87.3
		KLD	96.4	94.7	92.8	91.0

50 documents per author—KLD-based attribution method achieves an accuracy of 76.2% for 5-class classification, which is 5% higher than Bayesian networks. Approximately 6% improvement over Bayesian networks is achieved when large training data is used for 5-class AA as well. Both improvements have shown to be statistically significant with p-values less than 0.001 on a paired t-test We also observed that the smaller values of μ are the more effective, demonstrating that the influence of the background model should be kept relatively low.

We then run the corresponding experiments on the Gutenberg data, also as shown in Table 4.7; the method shows higher effectiveness with Gutenberg data. The trends observed are the same as on the AP data, illustrating that the method is robust between collections.

4.6.3 KLD as a Classifier for Text Categorization

In order to determine the suitability of KLD classification for other types of classification tasks, we use the Reuters-21578 collection, one of the standard benchmarks for text categorization, to test topic-based classification using KLD.

Reuters-21578. These documents are from the Reuters newswire in 1987, and have been used as a benchmark for general text categorization tasks. There are 21,578 documents in this collection. We use the Modapte split [Lewis et al., 2004] to group documents for training and testing, which is the one of the most widely used split methods.[1] The top eight categories are selected as the target classes; these are *acq*, *crude*, *earn*, *grain*, *interest*, *money-fx*, *ship*, and *trade*. In the Reuters-21578 data collection, documents are often assigned to more than one category. (This is a contrast to AA, in which each document has only one class.) In our experiment, we choose the first category as the labelled class, as it is the main category for that document. In common with standard topic classification approaches we used all document terms after stemming and stopping as the classification features.

In these experiments—we do not claim to have thoroughly explored the application of KLD to general categorization—we test n-class classification, where $n = 8$, both with and without stemming; results are in Table 4.8. We compare the KLD classification and SVMs using 1-against-n evaluation, in terms of precision, recall, and overall accuracy. Accuracy measures the number of documents correctly classified. Thus for any given category, it is calculated as the total number of documents correctly classified as belonging to that category, plus the total number of documents correctly classified as not belonging to that category, divided by the total number of documents classified. Results are shown in Table 4.9. KLD classification consistently achieves higher recall than SVMs, but with lower precision and accuracy. We conclude that KLD classification is a plausible method for general text categorization, but that further exploration is required to establish how it should best be used for this problem.

4.7 Chapter Summary

In this chapter we have proposed the use of relative entropy (KLD) as a method for identifying authorship of unattributed documents. Language models have formed the basis of this approach, which have been used for a recent series of developments in information retrieval, and have the advantage of simplicity and efficiency. Following information theoretic principles, we have shown that a basic measure of relative entropy, the Kullback-Leibler divergence,

[1] More details are presented in Chapter 2.

Table 4.8: Effectiveness (percentage of test documents correctly attributed) of KLD classification for general text categorization problem on the Reuters-21578 test collection.

categories	# of docs	KLD	KLD(stemmed)
top 8	train/test	$\lambda = 10^1$	$\lambda = 10^1$
acq	1482/668	92.51	92.81
crude	298/150	90.67	87.33
earn	2720/1048	97.61	96.95
grain	338/117	88.89	91.45
interest	172/80	58.75	68.75
money-fx	406/123	82.11	76.42
ship	137/54	59.26	75.93
trade	294/103	83.50	88.35

Table 4.9: Effectiveness (precision (pre), recall (rec), and accuracy (acc)) of KLD classification and SVM for general text categorization on the Reuters-21578 test collection. Best recall in each case is shown in **bold**.

categories	relevant/irrelevant	KLD($\lambda = 10^1$)	SVM
top 8 (1 vs. n)	(same train/test split)	rec/pre/acc	rec/pre/acc
acq	668/1675	**95.81**/93.70/96.97	94.01/96.32/97.27
crude	150/2193	**96.58**/62.95/96.24	69.33/91.23/97.61
earn	1048/1295	97.23/90.02/93.94	**98.19**/98.19/98.38
grain	117/2226	**99.15**/71.17/97.95	84.62/99.00/99.19
interest	80/2263	**92.50**/45.68/95.99	37.50/93.75/97.78
money-fx	123/2224	**95.12**/54.42/95.56	69.11/80.95/97.52
ship	54/2289	**85.19**/33.58/95.78	24.07/86.67/98.16
trade	103/2240	**93.20**/52.17/95.95	67.98/87.50/98.16

can be used for effective authorship attribution.

The machine learning methods evaluated in Chapter 2 are computationally expensive and, despite their sophistication, at their best can only equal the proposed KLD approach. Evaluation of the proposed KLD method was on two collections of different domains: newswire articles and English literature. Various smoothing techniques have been explored for approximation of the distributions of style markers, that is, the function words in our experiments. It turned out that the the query-centric smoothing process as used in IR is not always plausible or stable for effective authorship attribution. Therefore, we suggest that smoothing in AA should apply not only to in-document style markers but also to those absent from the test documents but pre-defined in the feature set. This smoothing process has shown to be more effective and reliable, in particular for discriminating between difficult author combinations. In addition, provided with a sufficient training documents, all smoothing methods in language models are effective for AA.

Importantly, we have compared our KLD-based AA method to the best baseline methods investigated in Chapter 3—SVMs and Bayesian networks—using the same style markers. We conclude that our KLD-based method is consistently more efficient and effective than Bayesian networks for both two-class and multi-class authorship attribution. It also outperforms SVMs with smaller training data in binary AA; however, with large training data, SVMs are slightly better, but the differences are tiny and not statistically significant. Our method is superior than SVMs for multi-class AA. Our KLD-based method can be directly apply to n-class AA, with any values of n in theory; however, it is not straightforward to apply SVMs for multi-class problems. Instead of making a direct distinction between n classes, SVMs usually convert an n-class problem to a number of n binary problems that are easier tasks. We therefore conclude that our KLD-based model is superior to existing approaches.

Chapter 5

Style Markers in Authorship Attribution

We have shown in Chapter 4 that our KLD-based approach can provide effective authorship attribution (AA). Provided with the most widely used style markers—that is, function words—the KLD-based approach has been shown to be better than the baseline methods in most cases, including methods such as SVMs and Bayesian networks; SVMs were slightly better when given a large number of training samples, however the difference was tiny. Importantly, function words are not the only possible choice of style markers; many other features have been proposed as style markers in previous AA research. However due to diversity of the experimental setup, the effectiveness of such feature types cannot be compared. Thus, it is of value to examine the discrimination power of various marker types.

In this chapter we investigate various types of style markers, determining which are the better choices for AA. Seven marker types are extracted and evaluated on the *AP7* corpus, by consistently applying the proposed KLD-based approach. Our results show that the richer or more sophisticated style markers do not necessarily lead to higher effectiveness of AA. Moreover, there is no single type of style marker that can satisfy all attribution scenarios. In this respect, we further propose three systems to take advantage of a combination of evidence, thus significantly improving the AA effectiveness.*

*This chapter incorporates work originally published by Zhao and Vines [2007].

5.1 Style Markers

Style markers, the key elements of effective AA, are features extracted from documents; these features are believed to be informative in reflecting the writing style of a particular author; and poor choices of style markers can lead directly to a failure of attribution. In linguistics, each sentence in a language, including English, has two levels of representation: a deep structure and a surface structure. The deep structure represents the semantic relations of words in the sentences, and is mapped on to the surface structure. Surface linguistic features describe the surface structure, and can be extracted without taking into account the semantic relations of a sentence; examples include topic words, function words, punctuation symbols, and some grammatical components such as part-of-speech tags. Deep linguistic features are extracted by analysis on the deep structure of texts, such as a syntax tree.[1]

In previous research, many types of linguistic features have been proposed as the style markers for AA. However there is no consensus on which style markers are superior to others. This incomparability is caused by differences in three facets: the collections used, the classification methodologies applied to the extracted markers, and the evaluation methods. Therefore in order to examine the discrimination power of various types of markers in AA, all three of these elements need to be held constant. This is to ensure that differences in the AA effectiveness are caused by the style markers only.

In the section below, we describe the seven types of style markers that are evaluated in this research. Four types of markers are extracted from the surface-level linguistic features and the others are deep-level linguistic features. Given a piece of sample text such as:

> *The widow she cried over me, and called me a poor lost lamb, and she called me a lot of other names, too, but she never meant no harm by it.*

we briefly explain the investigated style markers:

Function words (FW). These are words that have little semantic content of their own but are grammatically important, such as prepositions, conjunctions, and articles, etc. Function words, simple lexical features, are an obvious choice of feature for authorship attribution, as

[1] An example of a syntax tree is depicted in Figure 5.1.

they are independent of the content and may be a good indication of writing style. Many prior AA studies have employed such features for AA [Binongo, 2003; Burrows, 1987; Holmes, 1994; Holmes et al., 2001; Juola and Baayen, 2003]. We use FW to indicate function words in this chapter. The function words extracted from the above text are:

the over and a and a of other too but never no by it

Function words in a first order Markov chain (2GramFW). If considering the probabilities of occurrences of a particular function word conditional on the preceding function words, then bi-gram function words are extracted. In this way the bi-gram function words are sometimes known as first order Markov chain, notated as *2GramFW* in this research. The distribution can be measured by bi-gram language models rather than uni-gram language models.[2] After feature extraction, the representation of the original example text becomes:

the/over over/and and/a a/and and/a a/of of/other other/too too/but but/never never/no no/by by/it

Punctuation symbols. A related choice of feature is punctuation, though the discrimination power of a limited number of punctuation marks is low. They are rarely used merely by themselves, but may be complementary to other style markers. The extracted punctuation marks from our sample text are:

, , , .

Part-of-speech tags (POS). Alternatively, some researchers have explored usage of natural language processing (NLP) for AA [Baayen et al., 1996; Stamatatos et al., 2001]. We use the NLTK package to annotate documents with part-of-speech tags.[3] POS tags are lexical categories. These features are recognized and categorized by linguists, and each category can then be further classified according to the morphology. We apply NLTK to the entire AP data; a list of 183 tags[4] are extracted, forming the pre-defined feature set.

[2]For details please refer to Chapter 2.

[3]NLTK, a Natural Language ToolKit implemented in Python, is used in this research. The package is available from http://nltk.sourceforge.net/index.html.

[4]A list of extracted tags are provided in Appendix B.

Automated annotation requires learning. To annotate our corpus with POS tags, a tagger is trained by learning patterns from the pre-annotated *Brown* corpus. To choose a tagger for annotation, we trained both a uni-gram tagger and a bi-gram tagger. We compare the effectiveness of the taggers, by applying them to annotate a non-annotated version of *Brown* articles. The trained uni-gram tagger provides 87% accuracy, which is more effective than the bi-gram tagger by approximately 4%, and is thus chosen for the tagging process. However, the annotation is not perfect. It is intuitive that the tagger fails to correctly annotate words that are missing, or have insufficient occurrences in the training *Brown* corpus. These words may be tagged incorrectly, or remain *unknown*, tagged with *ZZ* in our case. After extracting POS tags while keeping punctuation marks, the original text is represented as:

> *AT NN PPS VBD IN PPO, CC VBD PPO AT JJ VBN ZZ, CC PPS VBD PPO AT NN IN AP NNS, QL, CC PPS RB VBD AT NN IN PPO*

in which, for example, *NN* is a noun and *AT* is an article.

Function words with POS tags (FW/POS). We propose this type of style marker based on the observation that a function word can play multiple grammatical roles in sentences. This is usually reflected by the cases where a function word can be annotated with different POS tags. We hypothesize that this kind of pattern can indicate writing habit, and therefore, we propose the style marker as a function word together with its POS for attribution purposes. The representation of the original text after feature extraction becomes:

> *the/AT over/IN and/CC a/AT and/CC a/AT of/IN other/AP too/QL but/CC never/RB no/AT by/IN it/PPO*

The above four marker types can be extracted from the surface structure of texts, by shallow linguistic parsing, that is, no syntactical analysis is involved. We also propose style markers based on the deep structure of texts. Taking the sample sentence, after applying deep linguistic parsing the syntax tree can be represented as in Figure 5.1. In this thesis, the *Stanford Parser* is used; several versions of the parser have been released, the one we consistently used was distributed in March 2004.[5] It is one of the state-of-art parsers for extraction of deep

[5]The official website is http://nlp.stanford.edu/software/lex-parser.shtml.

```
(ROOT
(S------------------------------------------------------------S-Depth-1
  (NP
    (NP (DT The) (NN widow))
    (SBAR
      (S-----------------------------------------------------S-Depth-2
        (NP (PRP she))
        (VP
          (VP (VBD cried)
            (PRT (RP over))
            (NP (PRP me)))
          (, ,)
          (CC and)
          (VP (VBD called)
            (NP (PRP me))
            (NP (DT a) (JJ poor) (JJ lost) (NN lamb)))))))
  (, ,)
  (S (CC and)--------------------------------------------S-Depth-2
    (NP (PRP she))
    (VP (VBD called)
      (S---------------------------------------------------S-Depth-3
        (NP (PRP me))
        (NP
          (NP (DT a) (NN lot))
          (PP (IN of)
            (NP (JJ other) (NNS names)))))
      (, ,)
      (ADVP (RB too))))
  (, ,)
  (CC but)
  (S------------------------------------------------------S-Depth-2
    (NP (PRP she))
    (ADVP (RB never))
    (VP (VBD meant)
      (NP (DT no) (NN harm))
      (PP (IN by)
        (NP (PRP it)))))
  (. .)))
```

Figure 5.1: An example of the grammatical structure of a sentence, represented as a syntax tree.

linguistic features, the main ideas of which were proposed by Klein and Manning [2002]. As shown, each word in the sentence is located at the leaf of that tree, and annotated by POS tags. In addition to the POS tags, relationships between the tags are further analysed and annotated by non-leaf nodes in the syntax tree. For example, *NP* indicating a noun phrase and *VP* indicating a verb phrase. The computation of this kind of grammatical analysis is expensive, indicating that efficiency is an issue. Also, the effectiveness of this kind of parsing directly impacts the effectiveness of AA. However, it conveys information beyond the text itself and is certainly worth exploring.

In comparison with literature work, newswire stories are generally short; the average document length in the *AP7* collection is 724 terms. It is important to ensure that the proposed style markers have sufficient instances to be used for training. Overly complicated features are not plausible for attribution of relatively short documents, which are likely to have extremely high sparseness and result in bad training models for prediction. Based on the syntax tree, we propose alternative features as potential style markers.

Noun phrase (NP). By definition, a noun phrase is a phrase that can function as the subject or object of a verb. A noun phrase can be nested in another to constitute a longer noun phrase; a long noun phrase can be formed from several short ones. The hypothesis is that authors have different preference in use of noun phrases; some prefer short ones, others may favour longer ones. The noun phrases are annotated with *NP* in the tree, as shown in Figure 5.1. For instance, *the widow* is a noun phrase of length two, and *a poor lost lamb* is a noun phrase of length four; both of the two are simple noun phrases that are not nested. The phrase *a lot of other names* is a nested NP that contains the smaller noun phrase, *other names*. Authors, such as Henry James, who prefer constructing long sentences, are likely to use longer NPs compared to those authors favoring simplicity. Therefore, we record the length of NPs; in the case of a nested NP, only the length of a complete phrase is considered. The parser annotate nouns as a NP of length one, which are not considered, as in fact they are individual terms rather than phrases.

Function words at S-Depth (FW/SD and FW|SD). This type of feature is motivated by the observation that function words are particularly important in the construction of long

sentences. As shown in the syntax tree, the symbol "S" indicates a *S-segment* that can function as a sentence. The complete sentence is nested by several *S-segments*. We define the segment at *S-depth-1* as the complete sentence, and the segments at *S-depth-n* as smaller segments that can function as sentences, but are parts of the complete sentence. Similarly, segments at *S-depth-n* are parts of segments at *S-depth-(n-1)*. We record the function words, as well as the corresponding *S-depth* information as a potential writing pattern. With the sample text and the corresponding syntax tree, the features that can be extracted are:

S-depth-1: *the but*
S-depth-2: *over and a and too never no by it*
S-depth-3: *a of other*

If considering only function words as style markers, we observe that *and* occurs twice, and *a* occurs twice. While by considering *S-depth*, we have more information about where the function words tend to appear: *and* is at *S-depth-2*, and *a* at *S-depth-2* or *S-depth-3*. Based on such kinds of information, we propose two features as the style markers. $FW|SD$ is measured by conditional probability, that is, the probability of occurrence of a function word conditional on S-depth, which takes probability of each possible S-depth into account, whereas FW/SD simply indicates a function words with its S-depth, and no dependency between FW and *S-depth* is taken into account.

5.2 First Results: Individual Marker Types

First we experiment with different forms of style markers individually. All results are comparable to those reported in Chapter 4, where type FW was used as the style marker. Both binary AA and multi-class AA are evaluated on the $AP7$ data, by applying the proposed KLD-based AA approach. For approximating the distribution of style markers, Dirichlet smoothing is used, which has shown to be the best in our early experiments; the smoothing parameter μ is tuned for the highest effectiveness. We compare the aforementioned seven types of style markers; four out of seven are extracted by shallow linguistic parsing: FW, POS, 2GramFW, and FW/POS; the other three are from deep linguistic parsing: NP, FW/SD, and FW|SD. The number of training documents varies from 25 to 600, for each attribution task with

each of the seven marker types. Results are presented in Table 5.1; each accuracy value reported is averaged from the discrimination between all possible author combinations, that is, 21 combinations for 2-Class AA (C_7^2) as well as 5-Class AA (C_7^5), and 35 combinations for 3-Class AA (C_7^3) as well as 4-Class AA (C_7^4).

As shown in Table 5.1, style markers from deep linguistic parsing are generally worse than those from shallow linguistic parsing, in particular with the more difficult multi-class AA. Amongst all deep linguistic features, NP is particularly poor; clear failure is observed even with the simplest binary AA, where the effectiveness is only slightly better than random. Another observation is that the deep-level linguistic features require more training data. The effectiveness is poor when only limited numbers of training samples are available, especially with multi-class AA. Considering 5-Class AA with 25 training samples, for example, FW|SD produces only 57.7% accuracy, which is 17% lower than the best performance achieved by POS tags. However when the volume of training data is increased to 600, there is only approximately 1% difference between these two types of style markers. Also, by increasing the number of training documents from 25 to 600, POS tags only improve the 5-class AA effectiveness by around 4%, however the improvement using deep linguistic features is more significant, by more than 20% by FW|SD for example.

The deep linguistic features are expensive to compute, but do not guarantee an improvement on the attribution effectiveness. In a typical scenario, an author does not usually have a large volume of writing material available. In this case, there will be insufficient data for good modelling with deep linguistic features, and thus, the learned model is likely to provide poor predictions. In this respect, it is not always plausible to extract deep linguistic features for AA purposes. An issue we explore is whether the effectiveness of AA can be significantly improved without extracting sophisticated linguistic features—that is, using style markers based on shallow linguistic features only.

The table 5.1 shows that, the four types of shallow linguistic features used in our experiments are similar in terms of the overall accuracy. While some methods are slightly better when using different volumes of training data, no one feature set provides a clear advantage, although the method *2GramFW* is nearly always worse than the others. However, although the results for each method averaged over a set of classification tasks are similar, further

Table 5.1: Effectiveness as a percentage of successful attributions for each type of style marker, by applying KLD classification model for authorship attribution. Evaluations are reported on both binary AA and multi-class AA, up to five classes (the highest effectiveness is shown in **bold** *in each case). The collection used is AP7.*

		Number of Training Documents					
n-Class	Marker Types	25	50	100	200	400	600
2-Class	FW	**88.2**	**89.9**	**91.2**	**91.8**	92.2	92.1
	POS	87.9	87.0	89.5	90.0	89.6	90.2
	2GramFW	82.0	83.7	87.1	89.8	92.6	**93.4**
	FW/POS	87.6	88.4	91.0	**91.8**	**92.4**	92.5
	NP	60.1	64.3	64.3	64.4	65.0	64.7
	FW/SD	84.6	86.4	89.7	90.9	91.6	91.9
	FW|SD	81.1	83.3	85.6	88.2	89.6	90.5
3-Class	FW	81.8	**83.9**	**86.6**	**87.5**	**88.3**	**88.1**
	POS	**82.1**	80.2	83.7	84.3	84.1	84.9
	2GramFW	72.0	75.1	79.5	84.1	88.1	89.4
	PW/POS	81.1	82.0	85.9	87.0	88.1	**88.1**
	NP	40.1	47.8	47.8	48.3	48.8	48.5
	FW/SD	73.9	78.7	83.2	85.1	86.3	86.9
	FW|SD	70.0	73.9	77.4	81.2	83.7	84.9
4-Class	FW	77.1	**79.9**	**82.7**	**83.8**	84.9	84.7
	POS	**78.0**	75.5	79.8	80.4	80.4	81.2
	2GramFW	64.9	69.6	74.3	79.9	84.8	**86.5**
	FW/POS	76.7	77.5	82.2	83.5	**85.2**	84.9
	NP	34.8	37.9	38.3	39.1	39.4	39.0
	FW/SD	70.2	73.2	78.3	80.9	82.5	83.4
	FW|SD	62.9	67.3	71.5	76.0	79.3	80.7
5-Class	FW	73.5	**76.7**	**79.7**	**80.9**	82.2	82.0
	POS	**74.8**	72.0	76.9	77.5	77.5	78.3
	2GramFW	59.6	65.8	70.5	76.6	82.1	**84.3**
	PW/POS	73.2	74.0	79.2	80.8	**82.9**	82.5
	NP	28.7	31.2	32.0	33.1	33.3	32.9
	FW/SD	62.6	69.1	74.5	77.5	79.5	80.7
	FW|SD	57.7	62.2	67.0	71.9	75.8	77.2

investigation has shown considerable differences on individual classification tasks. Different methods seem to provide more effective discrimination for some author pairs but not others. We conjecture that some authors have stylistic habits that result in the use of one form of style marker in a manner that is more distinctive than others. Figure 5.2 and Figure 5.3 show the results of using each set of features on an author by author basis, for 3-class AA and 4-class AA. The x-axis represents each author combination and the y-axis shows the corresponding attribution accuracy achieved by each of the four types of style markers. Clearly, no one method performs best for all author combinations, though POS is consistently poor.

However the strength of different kinds of style markers varies significantly, given different scenarios and attribution tasks. Therefore we are motivated to propose three authorship attribution systems that combine style markers in different ways to achieve better performance: a *model voting system*, a *two-stage model prediction system*, and an *additive modelling system*.

5.3 Authorship Attribution via Combination of Evidence

A straightforward way of combining features is to build a single model using multiple feature types, but little success was observed in our experiments. It seems that the distinguishing features peculiar to given author combinations have a greater tendency to be overwhelmed by other features. In the following section we describe the proposed three systems for authorship attribution and standard evaluation metrics. The *model voting system* provides a principled way of integrating existing approaches with little modification on their own. The *two-stage model prediction system* is much less expensive in terms of computational cost. The *additive modelling system* performs the best amongst the three, but with slightly higher computational cost. To formulate the three systems mathematically, we revisit some of the notations and basic formulae used in Chapter 4:

- m is a style marker selected from several possible style markers. Each document is then represented as a sequence of style markers. We have four choices for m.
- a indicates a particular author.

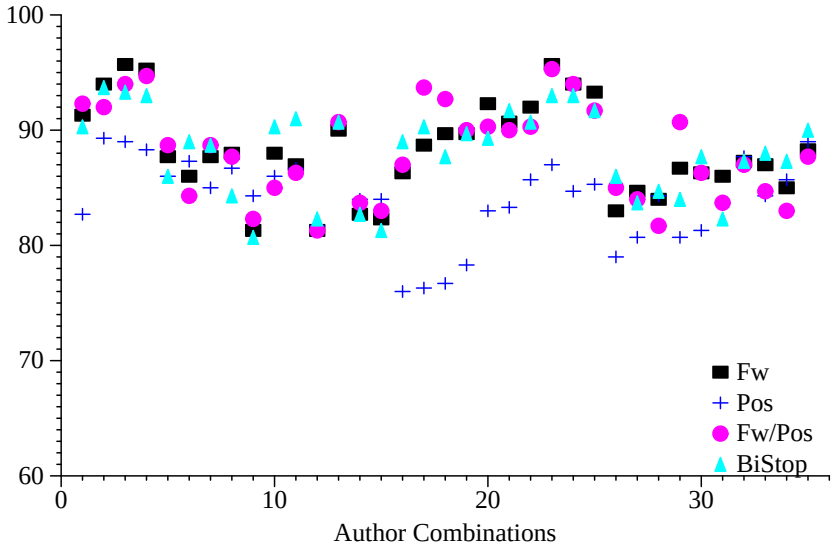


Figure 5.2: An example of **3-class** *authorship attribution. The results are obtained by four types of style markers based on shallow linguistic features. 400 documents per author are selected for training. The x-axis represents each author combination and the y-axis shows the corresponding attribution accuracy achieved by each of the four types of style markers.*

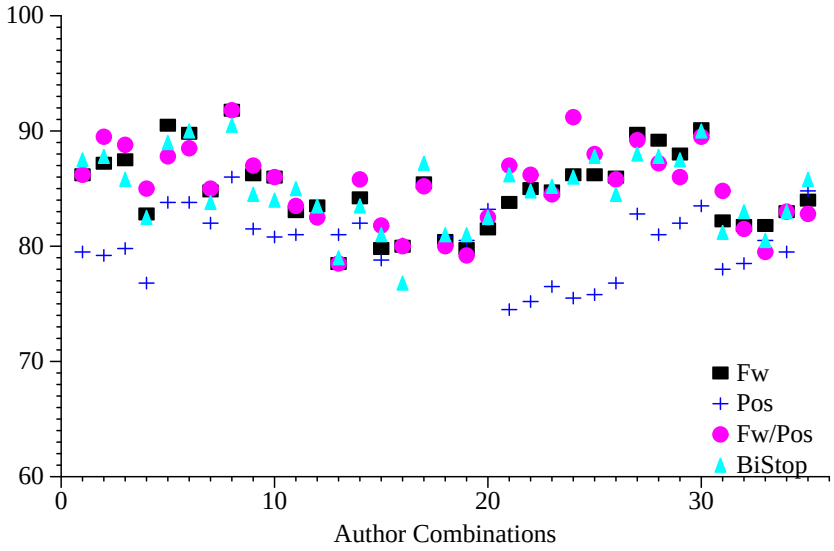


Figure 5.3: An example of **4-class** *authorship attribution. The results are obtained by four types of style markers based on shallow linguistic features. 400 documents per author are selected for training. The x-axis represents each author combination and the y-axis shows the corresponding attribution accuracy achieved by each of the four types of style markers.*

- d refers to a document, and $|d|$ is the length of the document after extraction of style markers.
- d^a refers to a set of training documents by the author a.
- $\hat{\theta}_{d|m}$ is an estimated document model based on the marker type m.
- $\hat{\theta}_{a|m}$ is an estimated author model based on the marker type m.
- Distributions of style markers in a document d, including *FW*, *POS*, *FW/POS*, and *2GramFW*, are measured and combined with Dirichlet smoothing:

$$\hat{p}_{\mu,\hat{\theta}_{d|m}}(f_i) = \frac{f_{i,d}}{\mu+|d|} + \frac{\mu}{\mu+|d|}\, p_{\hat{\theta}_{B|m}}(f_i)$$

  where μ is a smoothing parameter.

- The final classifier based on the estimated probability distributions is:

$$A = argmin_a \left(KLD\left(\hat{\theta}_{d|m}||\hat{\theta}_{a|m}\right)\right)$$

$$KLD\left(\hat{\theta}_{d|m}||\hat{\theta}_{a|m}\right) = \sum_{f_i \in F}\left[\left(\frac{f_{i,d}}{\mu+|d|} + \frac{\mu}{\mu+|d|}p_{\hat{\theta}_{B|m}}(f_i)\right) \times \log_2 \frac{\frac{f_{i,d}}{\mu+|d|} + \frac{\mu}{\mu+|d|}p_{\hat{\theta}_{B|m}}(f_i)}{\frac{f_{i,d^a}}{\mu+|d^a|} + \frac{\mu}{\mu+|d^a|}p_{\hat{\theta}_{B|m}}(f_i)}\right]$$

5.3.1 Model Voting System

The idea of the model voting system is simple: in order to attribute a document to a certain author, different types of style markers are modelled separately. Each model arrives at an explicit attribution result, a vote in other words, for the document to be identified. The author who gets the most votes wins. The framework of the voting system is depicted in Figure 5.4.

Given l different marker types, each of them is modelled to make a vote to one of the k potential authors. For a document d' to be attributed, the KLD is computed between the document d' and each author individually, based on each one of the l models of style markers. Therefore $l \times k$ KLD values are computed for l votes. The model voting process given one marker type can be formulated as:

$$V_{d'} = \forall_m \left[argmin_{\,a}\left(KLD\left(\hat{\theta}_{d'|m} \parallel \hat{\theta}_{a|m}\right)\right)\right] \tag{5.1}$$

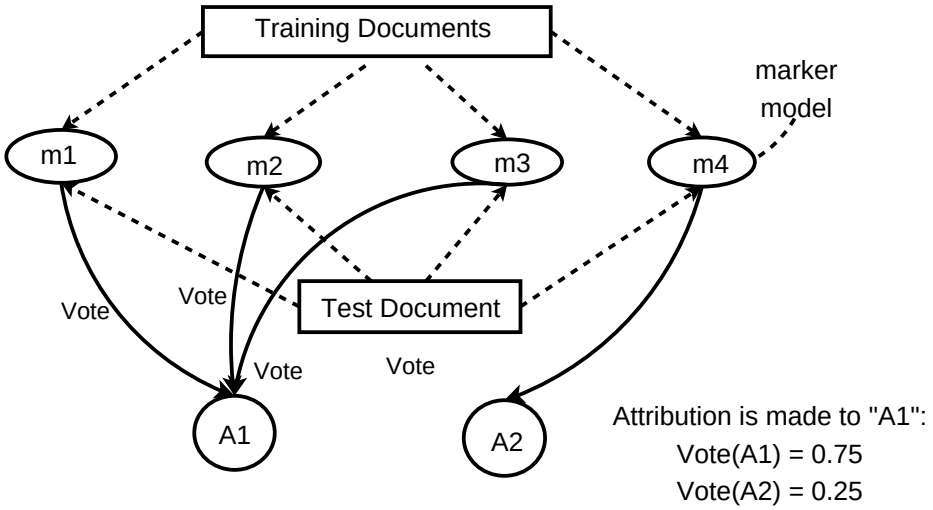


Figure 5.4: Framework for the model voting system.

The calculation of Equation 5.1 indicats that each one of the l types of style markers votes to a particular author. Eventually a set of votes $V_{d'}$ for the document d' with missing authorship is generated; each element in the set is the number of votes that each of the k potential authors gets from the system:

$$V_{d'} = \left(v_{d'}^{a_1}, \ldots, v_{d'}^{a_k}\right)$$

where values of the elements in $V_{d'}$ are in the range $[0, l]$, and $l = \sum_a v_{d'}^a$. It is possible that some authors get no votes.

In the model voting system, the more votes an author gets, the more likely it is that document d' was written by this author. The author A with the most votes wins and is attributed to the document d', that is:

$$A \quad = \quad argmax_a \, v_{d'}^a \quad \text{where} \quad a = (a_1, \ldots, a_k) \tag{5.2}$$

However in a formal way, a threshold δ is set for a stricter attribution. The document d' is attributed to the author A, if and only if A:

- satisfies the Equation 5.2, and
- $v_{d'}^A / l \geqslant \delta$, where l is the number of marker types.

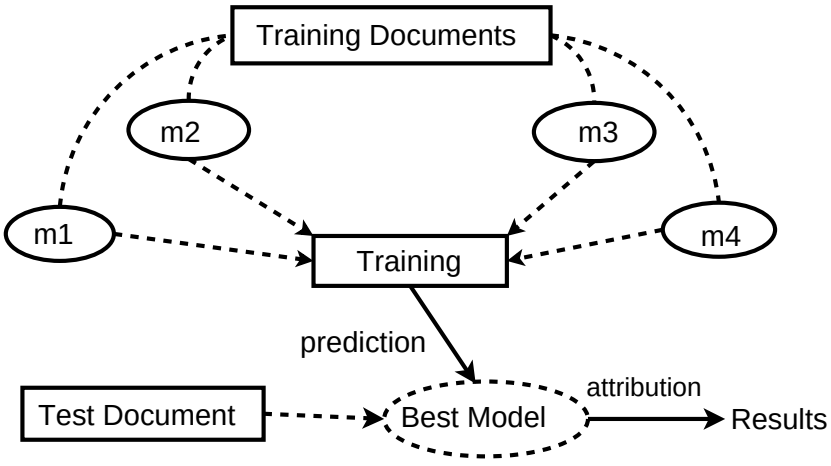


Figure 5.5: Framework of the two-stage model prediction system.

The threshold δ can be any real number in the range $(0, 1]$. Choices of δ depend on the number of potential authors and the number of marker types. The value of δ sets the strictness or reliability of a model voting system: the bigger the value of δ, the stricter the attribution. Using four marker types for binary AA as an example, alternative values for δ cab be $\{1, 0.75, 0.5\}$.

The methodology of the model voting system also provides a principled way of merging various existing AA approaches with little modification on the technique itself; only decisions made by these methods are required as input for the model voting system.

5.3.2 Two-Stage Model Prediction System

In the model voting system, both training and individual testing documents have to be fully modelled with each of the marker types available. In contrast we propose a two-stage model prediction system that is less expensive in terms of computational cost compared to the model voting system.

The outline of the two-stage model prediction system is shown in Figure 5.5. There are two steps involved in the two-stage model prediction system: prediction and attribution. In the prediction stage, the training documents for each author are split into two groups; all models

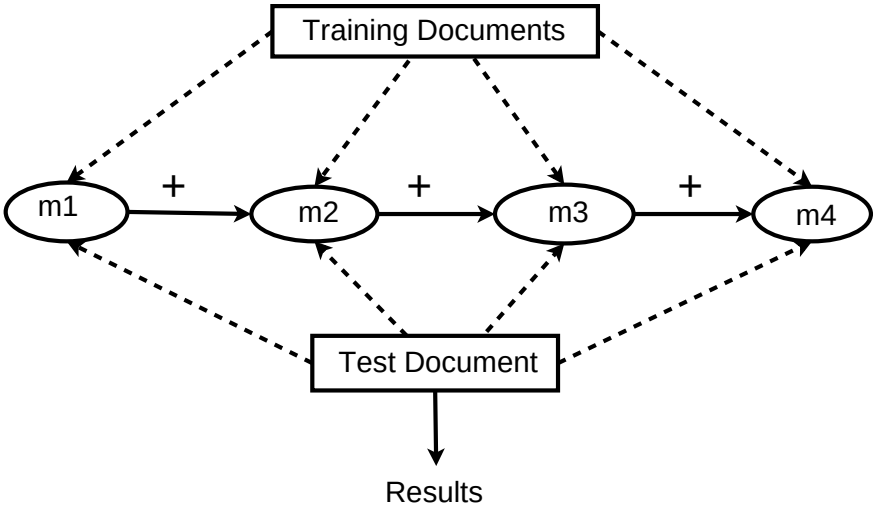


Figure 5.6: Framework of the additive modelling system.

are built for each author separately, based on documents in the first group; these marker models are then used to attribute documents in the second group. Thus, the effectiveness of various types of style markers can be compared in the prediction stage, and the one with the highest accuracy is selected. In the next attribution stage, the actual test documents with unknown authorship are identified using the selected style markers only. In this sense, the two-stage model prediction system has less computational cost than the model voting system, as only one out of the l marker types is modelled in the attribution stage.

5.3.3 Additive Modelling System

The additive modelling system uses the same modeling process as the voting system: for both training and test samples, full modelling is required for each author and each type of style marker individually. However the additive modelling system is different from the voting system in that different rules are used to make the final attribution. Rather than giving a unit vote by each feature model, it calculates the "size" of the vote, in accordance with the magnitude of divergence that each feature model produces. The outline of the additive modelling system is shown in Figure 5.6.

Given a scenario where three models out of four might slightly favor author A for an

unknown document, while only one model may favor author B, the model voting system would straightforwardly attribute the document with author A. However the additive system may select author B for the unknown document, if the divergence measured by one feature model is small enough. The additive system is formulated as:

$$A = argmin_a \left[\sum_m \alpha_l \left(KLD \left(\hat{\theta}_{d'|m} \parallel \hat{\theta}_{a|m} \right) \right) \right] \quad \text{where} \quad \textstyle\sum_l \alpha_l = 1 \tag{5.3}$$

where α_l is an adjustable parameter corresponding to each marker type m. In a simplest case, α_l can be set as a constant; or α_l can be computed by learning algorithms. The value of α_l indicates the significance of the marker type m for attribution in the additive system. It is also feasible to use the prediction stage of the two-stage prediction system in the additive system for weighting α_l, as the effectiveness of each marker type can be evaluated.

5.4 Experiments and Results

The aim of this research is to examine whether the three proposed AA systems can improve the attribution effectiveness. In order to draw comparable results, the experimental setup is kept consistent, using $AP7$ data with the same splitting method. We evaluate the three AA systems using multiple types of style markers. As mentioned, the deep-level linguistic features are avoided due to the high computational cost and low effectiveness. Four types of style markers based on shallow linguistic parsing are used. Therefore, in all experiments described below, $l = 4$.

Model voting system. We implement the model voting system as described in Figure 5.4. An appropriate value of threshold δ depends on the number of marker types that are available to the system, as well as the number of authors to be differentiated. For example, in the case of binary AA, the threshold can be set to any value greater than 0.5; whereas for n-class AA, the threshold can be set greater than $1/n$. In the voting system, if more than two authors get the most votes then the document with unknown authorship remains unattributed. Therefore the number of marker types required for voting is preferred to be much larger than the number of potential authors that is, $l > n$; for example, a voting system with three marker types

Table 5.2: The effectiveness of the voting system on binary authorship attribution. Results are compared to the best results of using individual feature.

		$\|M\|/\delta$			
Number of training samples	Evaluation	4/1	4/0.75	3/1	3/0.67
25	baseline		88.2		
	Rec_{voting}	70.9	86.0	79.6	89.2
	Pre_{voting}	**95.4**	91.5	93.9	89.2
100	baseline		91.7		
	Rec_{voting}	75.7	90.2	83.6	92.2
	Pre_{voting}	**97.9**	94.7	96.0	92.2
400	baseline		92.8		
	Rec_{voting}	81.1	92.1	83.6	92.9
	Pre_{voting}	**98.7**	95.2	95.7	92.9

for 4-class AA is unlikely to be plausible. Given four types of available style markers in our experiments, we evaluate binary AA for illustration.

We implement the model voting system with both three and four types of available style markers. When voting with four types of markers, the threshold δ can be set at $\delta = 0.75$ for instance, that is, the attribution can be made to a document d' if and only if $v^a_{d'} = 4$ or $v^a_{d'} = 3$. Moreover when δ is increased to 1, only documents with $v^a_{d'} = 4$ can be attributed. Otherwise the document remains unattributed. The threshold can be set in a similar way when voting with three marker types. Table 5.2 presents results of using different values of δ in both cases. The scalability is also examined by varying the number of training documents; results are reported from 25 training documents up to 400. The notation Rec_{voting} in the table shows the percentage of total number of test documents that are attributed with correct authorship, and is somewhat akin to a recall measure. In use of the model voting system, some documents remain unattributed if they do not meet the threshold. Therefore we also calculate the values of Pre_{voting}; it shows the number of documents that are correctly attributed with authorship as a percentage of those for which attribution decisions can be made (i.e. that meet the threshold, and is somewhat akin to a measure of precision). It can be seen that, by

using the model voting system with an appropriate threshold, we can have a higher degree of confidence that the attributed documents have been done so correctly—up to 98.7%—using four marker types with 400 training samples per author. Even with smaller numbers of training documents, or lower thresholds, the values of Pre_{voting} are also much higher than the baseline results.

In the area of AA, researchers have proposed many ways to attribute authorship to texts [Baayen et al., 1996; Burrows, 2002; Diederich et al., 2003; Holmes et al., 2001; Koppel and Schler, 2003; Sarkar et al., 2005]. However the open challenge is the lack of benchmarks and consistent experimental setup, which has led difficulties to judge these approaches. The model voting system is a plausible alternative to make use of existing approaches as far as possible. There is no need to modify the approach itself, instead the output from these approaches can be used as the input to the model voting system for a more reliable attribution. However, the drawback of this type of AA system is that, it requires a certain amount of valid votes, that is, the value of l should be reasonably large, in particular for multi-class AA.

Two-stage model prediction system. In this experiment we implement the two-stage model prediction system as shown in Figure 5.5. In order to achieve accurate predictions, a sufficient number of documents should be provided in the first prediction stage. Therefore 400 training samples are used; however these are divided into two sets. As described in Section 5.3.2, the first set is used to model each of the marker types, and the second set is used to predict the best feature model to be used for that author combination. The predicted marker types for each individual author combinations are shown in Figure 5.7, illustrating the 3-class AA in accordance to the baseline results shown in Figure 5.2.

We experiment with both binary AA and multi-class AA, up to five classes; the effectiveness is reported in Table 5.3, which is comparable to the baseline presented in Table 5.1. It is observed that nine correct predictions are successfully made out of the 21 binary attribution tasks. Also 21 and 19 out of the 35 attribution tasks are predicted with the best marker type, for respective 3-class AA and 4-class AA. In contrast, the likelihood of any individual type of style marker giving the best result for a given author combination is always lower than if the prediction system is used. In cases where the best system was not chosen, the

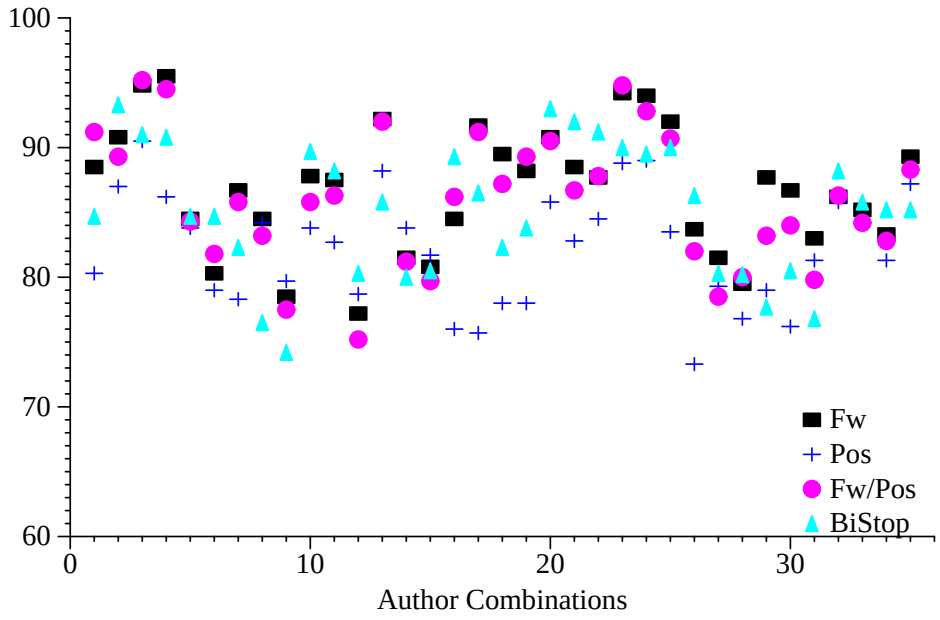


Figure 5.7: An example of predicting the best feature model for **3-class** *authorship attribution. The prediction for each author combination is achieved in the prediction stage by the two-stage model prediction system. The x-axis represents each author combination and the y-axis shows the corresponding attribution accuracy achieved by each of the four types of style markers.*

second best system was usually chosen and in these instances there was usually not a great deal of difference between the top systems. Also, the prediction system avoids choosing a marker type that is particularly bad for a given author combination.

A total of 112 attribution tasks are evaluated, from 2-class AA to 5-class AA, given 7 authors in total ($C_7^2 + C_7^3 + C_7^4 + C_7^5$). The likelihood for a single type of style markers to be the best choice of the task is only 35.7% (40/112) in our experiments, whereas, by model prediction, the likelihood can be increased to 53.6% (60/112). Approximately 18% improvement is achieved in terms of the probability of picking up the best feature model for a particular attribution task. On the other hand, the overall attribution accuracies are also improved by the two-stage model prediction system. On average around 1.5% improvement is achieved, shown as the last column in Table 5.3. Additionally the two-stage model prediction system is less expensive, since only one type of style marker needs to be modelled for individual test documents with unknown authorship, rather than all types. However in cases where only

Table 5.3: The effectiveness of using the two-stage model prediction system for 2, 3, 4, and 5-class authorship attribution tasks.

	M_c (# of combinations)						
N Class	FW	POS	FW/POS	2GramFW	M_c(pre)	Acc.	Acc.(Pre)
2 (/21)	6	4	4	7	**9**	92.3	**93.5**
3 (/35)	10	4	9	12	**21**	87.5	**88.8**
4 (/35)	11	3	16	9	**19**	83.8	**85.2**
5 (/21)	5	1	**12**	3	11	80.9	**82.4**
overall (/112)	33	12	40	32	**60**		

limited numbers of training samples are available, it is not ideal to use the two-stage model prediction, as the predictions based on insufficient training data are usually not reliable.

Additive system. Finally we implement the additive system as depicted in Figure 5.6. We evaluate the system by varying the number of available marker types and the number of available training samples that are provided to the additive system. Given four types of features in total, the numbers of combinations are C_4^2, C_4^3, and C_4^4. We effectively sum the divergences produced by each feature model (by setting a uniform value for all $\alpha = 1/l$), however it is possible that in a more sophisticated system we may also assign a different weight α_l to each of the marker models. The comprehensive results in effectiveness are presented in the Tables 5.4 and 5.5. In these tables, a "+" symbol indicates the combination of the selected types of style markers; a "−" symbol indicates the exclusion of that marker type, but using the rest of $l-1$ types of style markers.

It can be seen that the additive system gives the best effectiveness amongst all three proposed systems, however with slightly higher computational cost. The best result is achieved by modelling all four types of features with sufficient training data, that is, 400 training documents per author. The effectiveness of the most difficult 5 class AA is achieved at 90%, approximately a 7% improvement in comparison with the best baseline result achieved when using any of the individual marker types.

In addition, the choice of style markers affects the effectiveness of the additive system.

Table 5.4: The comparison between additive system on **two** *marker types, and the best results from use of individual types of style markers (also presented in Table 5.1). Give a number of documents for training, the best performance is shown in* **bold** *in each case—from 2- to 5- authors.*

Number of		N-class AA			
Training samples	$M_{selected}$	2	3	4	5
25	FW+POS	**90.3**	**85.4**	**81.7**	**78.6**
	FW+FW/POS	88.4	82.2	77.8	74.3
	FW+2GramFW	87.3	80.0	74.6	70.2
	POS+FW/POS	89.5	84.2	80.4	77.3
	POS+2GramFW	86.2	78.6	73.0	68.5
	FW/POS+2GramFW	87.3	80.6	75.6	71.6
	Best one	88.2	82.1	78.0	74.8
100	FW+POS	**92.6**	88.3	85.3	**82.9**
	FW+FW/POS	91.6	86.5	82.8	79.8
	FW+2GramFW	92.2	86.9	82.8	79.5
	POS+FW/POS	92.5	**88.4**	**85.4**	**82.9**
	POS+2GramFW	91.3	85.4	81.2	77.8
	FW/POS+2GramFW	91.8	86.5	82.7	79.7
	Best one	91.2	86.6	82.7	79.7
400	FW+POS	92.9	88.6	85.0	82.9
	FW+FW/POS	92.7	88.3	85.1	82.6
	FW+2GramFW	95.0	91.9	89.5	87.6
	POS+FW/POS	92.9	89.0	86.2	83.9
	POS+2GramFW	95.1	92.1	89.7	**89.6**
	FW/POS+2GramFW	**95.2**	**92.4**	**90.2**	88.4
	Best one	92.4	88.3	85.2	82.9

Table 5.5: The comparison between additive system on more than **two** *marker types, and the best results from use of individual types of style markers (also presented in Table 5.1). Give a number of documents for training, the best performance is shown in* **bold** *in each case—from 2- to 5- authors.*

Number of		N-class AA			
Training samples	$M_{selected}$	2	3	4	5
25	-FW	88.8	83.2	79.0	75.7
	-POS	88.8	82.8	78.4	74.8
	-FW/POS	89.1	83.3	78.9	75.3
	-2GramFW	**89.4**	**84.0**	**80.0**	**76.7**
	all	**89.4**	83.9	79.8	76.5
	Best one	88.2	82.1	78.0	74.8
100	-FW	92.7	88.1	84.7	81.9
	-POS	92.6	87.8	84.3	81.5
	-FW/POS	93.0	88.2	84.5	81.5
	-2GramFW	92.5	**88.8**	85.2	**82.8**
	all	**93.1**	88.7	**85.4**	82.7
	Best one	91.2	86.6	82.7	79.7
400	-FW	95.5	92.7	90.7	89.0
	-POS	95.5	92.8	90.8	89.3
	-FW/POS	**95.7**	93.2	91.2	89.6
	-2GramFW	93.1	89.0	86.0	83.5
	all	**95.7**	**93.2**	**91.5**	**90.0**
	Best one	92.4	88.3	85.2	82.9

Table 5.6: Significance test between the best baseline results and the best additive modeling results across all attribution tasks, using 400 documents for training (the confidence level is set to be 0.05).

	2	3	4	5
p-value	0.018	$1.291e^{-7}$	$2.053e^{-13}$	$5.177e^{-13}$
Significantly better	Y	Y	Y	Y

Given 400 training samples for instance, the combination of *FW/POS* and *2GramFW* gives the best results amongst all feature model pairs. While using all types of features except *FW/POS* gives the best results of any models that use three types. Other combinations achieve slightly lower accuracies, although still better than the baseline results. Table 5.6 shows the results of significance tests between the performance of the best baseline system from Table 5.1, and the best additive system using all feature models. The additive system performs significantly better, especially with the harder multi-class attribution tasks, with which the p-values are extremely small. It is likely that a better weighting scheme for estimating α_l can produce better results than what we report here, however our main interest is to examine the methodology itself.

5.5 Chapter Summary

We have examined the choice of style markers for effective authorship attribution. Though many features have been used for AA in literature, they are not comparable due to the diversity of the experimental environment. Moreover, the reliability of the published conclusions on style markers is unclear, as most work is based on fairly limited data and specific authors.

One of the primary contributions in this chapter is that we have evaluated and compared the effectiveness of individual marker types under a consistent experimental environment. Seven types of style markers have been extracted, by applying both shallow linguistic parsing and deep linguistic parsing. We have shown in this chapter that there is no one type of style marker that always works the best for all attribution tasks. Interestingly, style markers formed from deep linguistic features were somewhat less effective than lexical features, and

are more expensive in terms of computational cost.

Rather than exploring more advanced or sophisticated style markers, an alternative way to improve AA accuracy is to take advantage of multiple types of simple marker types. Another major contribution is that we have provided guidance on how to effectively use multiple types of style markers for AA. While simply adding more features into a given model has been shown to have little success, three AA systems have been proposed to combine evidence. All systems have proved to be highly accurate, in particular the additive system; all of them have been shown to be more effective than any previous method based on an individual mark type.

The model voting system has been able to provide a principled way of integrating existing approaches with little modification while enhancing the attribution performance. The threshold in the voting system can be adjusted from a value that maximizes the total number of documents correctly attributed to one that maximizes the probability that documents attributed are done so correctly. On the other hand, the two-stage model prediction system is much less expensive in terms of computational cost, while providing with not only higher attribution accuracy but also better choices of style markers in relation to a particular attribution task. Amongst all the three systems, the additive system has performed the best; the improvement in effectiveness has shown to be statistically significant.

Although the results achieved are quite positive, the number of authors being involved so far is not great, only 7 at the most. Also, size of the collections is not particularly large, even though it is much bigger than those used in previous literature. In the next chapter, we propose authorship search (AS), which addresses the scalability issue in AA.

Chapter 6

Authorship Search

Scalability is a beneficial property for an authorship attribution (AA) system, which indicates the ability of the system to handle substantial volumes of data. The scalability of an AA approach can be evaluated in two dimensions: by increasing the number of authors to be distinguished, and by increasing the number of documents included in the collections. While the newly proposed KLD-based attribution method has been demonstrated to be successful in previous chapters in terms of effectiveness and computational cost, it has not been scaled up to more than 5,000 documents, or more than seven authors. Despite these limits, we note that this is a much larger collection than those used in prior AA research.

We introduce the new task of authorship search (AS), the purpose of which is to search for documents written by a particular author, rather than search on a particular subject or topic. Our novel AS system employs the principle of the KLD-based methodology, where the divergence is proposed as a ranking mechanism, inspired by the language models used in information retrieval. The collections used for evaluation of the AS system are much larger than those used previously, consisting of half a million documents in the largest case. Our results show that the AS system is reasonably effective at identifying documents that share authorship. Also, we demonstrate that the AS approach can be used for AA, which is substantially more scalable than state-of-art approaches in terms of the collection size and the number of candidate authors. It is also the first time that the feasibility of AA and AS

on large document collections is demonstrated.*

6.1 Motivation

In our review of the AA literature, we observed that none of the previous AA approaches have been scaled to larger document collections; most of the collections are small, in terms of either the number of disputed authors, or the number of documents. We have developed the biggest collection to date: *AP7*, which consists of 5,000 news articles of seven authors. Our KLD-based method has scaled well with this collection, giving high effectiveness on both binary AA and multi-class AA, outperforming previous methods. However, a typical real-world text collection often contains either a much larger number of documents or authors.

Inspired by information retrieval systems that are highly scalable to massive volumes of textual data, we propose the novel task of authorship search, with the aim of handling substantially larger numbers of documents and authors. The purpose of authorship search (AS) is to find the documents that appear to have been written by a particular author within large collections. In other words, given documents of known authorship, the task is to find other documents that are written by the same author. AS is related to authorship attribution (AA), but has not been previously investigated.

Both IR and AS systems are approaches to search, but they have substantial differences. Modern IR systems deal with large volumes of information, and attempt to satisfy a user information need, by taking a query as input and returning in response, a list of relevant documents. IR systems are concerned with topical similarity between queries and documents, and thus, relevance is judged in terms of the content of the retrieved documents. In contrast, an AS system is concerned with writing styles of documents, so that the relevance of a document to a query depends on the similarity of writing patterns extracted from the retrieved documents and the query documents . As we describe in detail later, AS differs from topic-based IR in several key respects: query constitution, indexing scheme, and query evaluation.

*This chapter incorporates work originally published by Zhao and Zobel [2007b].

6.2 Methods of Authorship Attribution

Authorship attribution approaches are classification techniques. A range of approaches have been proposed, as discussed in Chapter 2. Most of these classification methods are not directly applicable to search tasks. In a search system a query is evaluated by ranking the similarities measured between the query and each document individually in the collection. The result is returned as a list of top-ranked documents. In contrast to search, there is no document ranking required for classification-based AA; instead, an explicit decision is made for each unknown document individually. The process of AA starts with a certain number of training documents for modelling the writing style of a particular author, where the training samples are aggregated to be treated as a whole rather than as individual documents, meaning that document-by-document calculation is not involved. From another perspective, search is more concerned with the connection of individual documents to a query, while attribution is more concerned with shared properties that can be generalised from a group of documents rather than a single sample. To the best of our knowledge, AA techniques have not been applied to search problems. In this chapter we propose what we believe is the first style search mechanism—authorship search (AS).

6.3 Document Search

Retrieval systems have been developed for textual data [Baeza-Yates and Ribeiro-Neto, 1999; Zobel and Moffat, 2006] as well as on multimedia data [Lew et al., 2006] such as images [Swets and Weng, 1996; Weber and Mlivoncic, 2003], audio [Tseng, 1999; Suga et al., 2004], and video [Chua and Ruan, 1995; Chang et al., 1997; Gaughan et al., 2003]. However textual data is still dominant. Current text retrieval systems usually deal with large and heterogeneous collections and typically take as input queries of few words, returning in response a list of documents deemed most likely to be relevant.

In general, there are two types of queries: Boolean queries and ranked queries. Boolean queries are evaluated by using a set of operators such as OR, AND, and NOT. Documents for which the Boolean conditions are satisfied are evaluated as "True", and returned in the results; all these documents are regarded as equally relevant to the given query. In most current

search engines, retrieved documents are ranked by some measure of similarity, indicating how likely it is that a document is relevant to the query. However, the returned documents are not necessarily useful to a user [Brajnik et al., 1996; Cooper, 1973]. In this respect, the evaluation of such search approaches requires relevance judgement on retrieved documents in relation to the input queries. Relevance has been explored elsewhere in literature [Mizzaro, 1997; 1998; Schamber, 1994; Sormunen, 2002].

Search generally involves two stages: index term extraction, and a methodology of similarity computation that is applied to the extracted index terms. For documents in English, extraction of index terms involves text-preprocessing: separating texts into individual words, case-folding, stopping, and stemming. [Witten et al., 1999; Zobel and Moffat, 2006]. A variety of ranking mechanisms have been proposed for the computation of similarity between the documents and queries including the vector space model [Salton and McGill, 1984; Bookstein, 1982; Baeza-Yates and Ribeiro-Neto, 1999; Melucci, 2005; Salton and Buckley, 1988; Singhal and Salton, 1995], probabilistic models [Robertson et al., 1980; Jones et al., 2000; Amati and Rijsbergen, 2002], and language models [Ponte and Croft, 1998; Croft and Lafferty, 2003; Liu and Croft, 2004; Bai et al., 2005]. We briefly describe these models below.

6.3.1 Vector Space Model

The vector space model has been extensively used in document retrieval [Wong et al., 1987; Zobel and Moffat, 1998; Melucci, 2006], and has been applied to other research areas such as text categorization [Joachims, 1997], and document filtering [Soboroff and Nicholas, 2000]. In a vector space model, queries and documents are represented as vectors of n dimensions, where n is the size of the collection vocabulary. The estimated similarity between a document and a given query is defined as the closeness of a document vector and a query vector, where the closeness is measured by degree of the angle between these two vectors. Intuitively, the closer the two vectors, the smaller angle it is. Documents whose vectors have accute angles to the query vector are considered to be of similar subjects or topics to a query. The similarity is quantified by *cosine* measure; we derive the mathematical formulae of *cosine* measure for ranking documents in the following description.

Table 6.1 summarises the commonly used notations in similarity measure. By definition,

Table 6.1: The atomic components used in similarity measure in IR.

notation	meaning
C	A test collection
$\|C\|$	The total number of terms in C
T	The total number of distinct vocabulary terms
t	A particular term
d	A particular document
$\vec{d}$	The vector of the document d
$\|d\|$	The number of terms in the document d
q	A particular query
$\vec{q}$	The vector of the query q
$f_{d,t}$	The frequency of term t in the document d
$w_{d,t}$	The weight of term t in the document d
$f_{q,t}$	The frequency of term t in the query q
$w_{q,t}$	The weight of term t in the query q
f_t	The number of documents containing a particular term t
W_d	The weight of the document d
W_q	The weight of the the query q
W_A	The average document weight
k_1	Tuning parameter, usually set to 1.2 empirically
b	Tuning parameter, usually set to 0.75 empirically
k_3	Tuning parameter, usually set to ∞ empirically
$S_{q,d}$	The similarity between a query q and a document d

the cosine of the angle between two vectors can be determined by the dot product of the two vectors. Query terms missing in the document and document terms missing in the query do not contribute to the cosine value, and thus, we present the formula as:

$$\begin{aligned} cosine\left(\vec{q}, \vec{d}\right) &= \frac{\vec{q} \cdot \vec{d}}{W_q \times W_d} \\ &= \frac{\sum_{t \in q \cap d} (w_{q,t} \times w_{d,t})}{W_q \times W_d} \end{aligned} \tag{6.1}$$

The component $t \in q \cap d$ specifies that only terms occurring in both the query and the document are computed. For the cosine measure, smaller angles between the vectors result in larger cosine values. Using this measure, documents are ranked by cosine value in decreasing order.

The document term weight $w_{d,t}$ and the query term weight $w_{q,t}$ are defined separately. Some terms are likely to be more important in relation to a query in topic or subject matter than others, and it is natural to assign higher weight to such terms. A family of weighting schemes—that distinguish between the importance of a term in the query, and the importance of a query in the collection—are known as $tf \cdot idf$ weighting, where two types of frequencies are concerned: term frequency (tf) and inverse document frequency (idf). Term frequency, tf, is determined as an increasing function of the within-document frequency $f_{d,t}$ that is the number of times that a query term t occurs within a document d. Inverse document frequency, idf, is defined as a decreasing function of f_t that is the number of documents containing a query term t.

Many variations of functions have been proposed for tf and idf—a detailed discussion is provided by Zobel and Moffat [1998], where the ranking scheme varied according to the Q-expression notation introduced. A Q-expression consists of eight letters written in three groups; each group is separated by hyphens indicating the ways of term weighting in documents and queries, such as BB-ACB-BCA.[1] In their work, the cosine similarity measure was decomposed into five components, including eight combination of functions, nine term weighting schemes, two definitions of document-term and query-term weights, six definitions of relative term frequency for document, and five for the query. They tested several variants

[1]The exact meaning of each character and formulae can be found in the paper provided by Zobel and Moffat [1998].

of measures, exploring whether some formulations are superior to others. However the results suggest that there is no single scheme that outperforms the rest. Following the same convention as *Q-expression*, we implement two competitive ranking systems that were shown reasonably effective in their research—BB-ACB-BCA and BB-BCI-BCA—for style search. The weighting scheme of Q-expression BB-ACB-BCA, which is required for Equation 6.1 is:

$$\begin{aligned} w_t &= \log_e 1 + \frac{N}{f_t} \\ w_{d,t} &= 1 + \log_e f_{d,t} \\ w_{q,t} &= w_t \times (1 + \log_e f_{q,t}) \\ W_d &= \sqrt{\sum_{t \in d} w_{d,t}^2} \\ W_q &= 1 \end{aligned}$$

For the Q-expression BB-BCI-BCA, the selected term weighting scheme is:

$$\begin{aligned} w_{d,t} &= w_t \times (1 + \log_e f_{d,t}) \\ w_{q,t} &= w_t \times (1 + \log_e f_{q,t}) \\ W'_d &= (1 - s) + s \cdot \frac{W_d}{\text{avg}_d W_d} \\ W_d &= \sqrt{\sum_{t \in d} w_{d,t}^2} \\ W_q &= 1 \end{aligned}$$

where s is the slope, which is usually set to 0.7 in IR [Singhal et al., 1996], and is consistently used in our AS investigation, and w_t, w_q are as above. Note we set the weight of the query vector W_q as a constant as it has no effect on rankings generated by the similarity measure. To use such kinds of models for AS, we apply the style markers (introduced in Chapter 2) to index the collections. We conjecture that a similar assumption would apply to AS: given appropriate index terms (style markers) and ranking schemes, the distribution of style markers in the documents by an author should be similar.

6.3.2 Probabilistic Models

An alternative ranking mechanism is probabilistic IR models that can be used to derive estimates for the probability that a document is relevant to a query [Robertson and Jones,

1976; Robertson et al., 1980]. The higher the probability, the more likely it is that a document is relevant to that query. The Okapi BM25 measure is one of the most successful IR measures based on the probabilistic model [Robertson et al., 1992; Jones et al., 2000; Amati and Rijsbergen, 2002], in which an estimation of the $2-Poisson$ distribution is usually assumed. The notations we use for this measure are also in Table 6.1; the similarity function $S_{q,d}$ can be defined as:

$$\begin{aligned} S_{q,d} &= \sum_{t \in q} w_{q,t} \cdot w_{d,t} \\ w_{q,t} &= ln\left(\frac{N - f_t + 0.5}{f_t + 0.5}\right) \cdot \frac{(k_3+1)\cdot f_{q,t}}{k_3 + f_{q,t}} \\ w_{d,t} &= \frac{(k_1+1)\cdot f_{d,t}}{K_d + f_{d,t}} \\ K_d &= k_1\left((1-b) + b\frac{W_d}{W_A}\right) \end{aligned}$$

Note that if the adjustable parameter k_1 is set to 0, the effect of within-document frequency of a particular term t diminishes, meaning that the significance of a term t is not increased by the number of times it occurs in a document. In contrast, if k_1 is set to a large number, the term weight increases linearly. The constant b is used to adjust the effect of document length; document length is not taken into account when b is 0. The parameter k_3 is used to adjust the weights of re-occurring query terms. Whether such a model is suitable for AS is, intuitively, not clear, but given the success of BM25 in IR it is reasonable to consider use of BM25 for AS. In this chapter, we also implement a standard BM25 model, exploring whether it can be applied to AS.

6.3.3 Language Models

In IR, given a document d and a document model $\hat{\theta}_d$ that is inferred from d, language models are used to estimate the probability that the document model $\hat{\theta}_d$ could have generated the query q [Ponte and Croft, 1998]. This is also known as query-likelihood model. Smoothing techniques (see details in Chapter 4) are typically applied to assign non-zero probabilities for query terms missing in the documents [Zhai and Lafferty, 2004; Chen and Goodman, 1996; Hiemstra, 2002].

Although language models have elements that are counter-intuitive—suggesting, for example, that queries comprised of common words are more likely than queries comprised of words that are specific to a given document—they are currently a highly effective approach to query evaluation in IR. In this chapter a key contribution is exploration of whether language models are suitable for AS. A similar approach has been proposed in Chapter 4 for AA.

6.4 Style-based Authorship Search

In both authorship attribution (AA) and authorship search (AS), the underlying assumption is that there are patterns or characteristics of an author's writing that can be automatically generalised and be used to distinguish their works from those of others. In Chapter 4 we have shown that, given appropriate style markers, documents from different authors can be separated, regardless of topic.

A key difference between document search and authorship search is choice of index strategies. IR techniques make use of content-bearing words, while in AS it is necessary to identify style markers for indexing, as we show later. Another potential difference is choice of similarity measure. We propose relative entropy as the similarity measure for AS, which is inspired by the language models used in IR and motivated by its successful achievement on AA. We compare the entropy based ranking methodology—as we demonstrate later—against other well known IR similarity measures, including the vector space model and probabilistic models.

6.4.1 Indexing Strategy

Extraction of index terms is a key aspect of an AS system. In contrast to IR, where content-bearing words are indexed, in AS we intend to extract content-free but style-informative components from the original documents to be indexed.

Many style markers were evaluated in Chapter 5, however not all of these marker types are suitable for AS. As a search system, reasonable high efficiency is required. Marker types—those based on syntactic relations—are not plausible, as they are quite difficult and expensive to extract, indicating that the low efficiency would be an issue. Moveover, the documents that we used are news articles; they are generally short, having 724 terms on average; overly sophisticated style markers are unlikely to be successful due to the lack of in-document

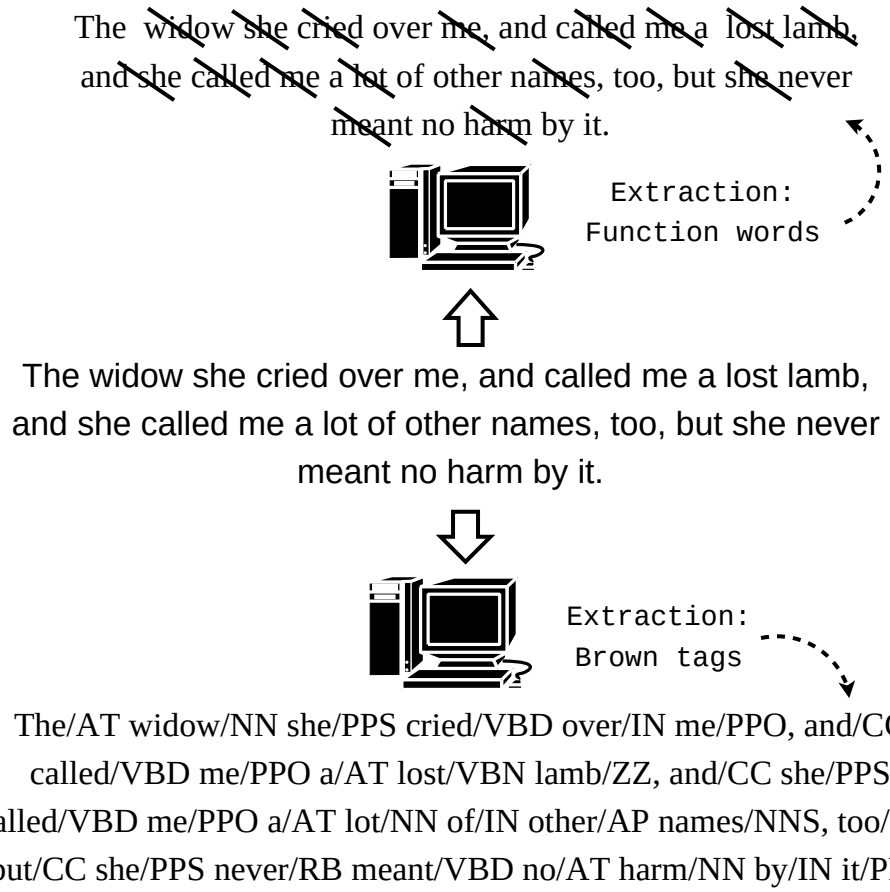


Figure 6.1: An example of index term extraction based on function words and POS tags.

observations for such markers. For our investigations, we choose to index collections with two types of style markers, function words and POS tags.

We have previously defined a list of function words in the Chapter 3; the entire *AP* sub-collection of TREC data was tagged by a POS uni-gram tagger in Chapter 5, from which list of 183 POS tags were extracted; we consistently use these markers for indexing. Figure 6.1 illustrates an example of the extraction of index components. The sample text is given in below:

> *The widow she cried over me, and called me a poor lost lamb, and she called me a lot of other names, too, but she never meant no harm by it.*

As shown, the function words extracted are "the over and a, and a of other, too, but never

no by it"; the POS tags are "AT NN PPS VBD IN PPO, CC VBD PPO AT JJ VBN ZZ, CC PPS VBD PPO AT NN IN AP NNS, QL, CC PPS RB VBD AT NN IN PPO", in which, for example, "NN" is a noun and "AT" is an article.

6.4.2 Entropy-based Similarity Measure for Authorship Search

We have proposed relative entropy—that is, Kullback-Leibler divergence—for classification in Chapter 4. Entropy measures the uncertainty of a random variable X, where, in this application, each $x \in X$ could be a token such as a word or other lexical feature, and the probability $p(x)$ is generated by the probability mass function of X. Kullback-leibler divergence (KLD) quantifies the dissimilarity between two distributions $P(X)$ and $Q(X)$ of the same random space:

$$KLD\left(P\left(X\right)||Q\left(X\right)\right) = \sum_{x \in X} p\left(x\right) \log_2 \frac{p\left(x\right)}{q\left(x\right)} \tag{6.2}$$

Now we propose a KLD-based technique for style-based AS. Table 6.2 presents a summary of the notations used in this section. A query q is supplied to an AS system, which has a language model $\hat{\theta}_q$. To rank documents in the collection, the similarity is measured between the query model $\hat{\theta}_q$ and the document model $\hat{\theta}_d$ of each document in the collection individually:

$$\begin{aligned} KLD\left(\hat{\theta}_q||\hat{\theta}_d\right) &= \sum_{t \in q} p\left(t|\hat{\theta}_q\right) \log_2 \frac{p\left(t|\hat{\theta}_q\right)}{p\left(t|\hat{\theta}_d\right)} \\ &= \sum_{t \in q} p\left(t|\hat{\theta}_q\right) \log_2 p\left(t|\hat{\theta}_q\right) - \sum_{t \in q} p\left(t|\hat{\theta}_q\right) \log_2 p\left(t|\hat{\theta}_d\right) \\ &= -\sum_{t \in q} p\left(t|\hat{\theta}_q\right) \log_2 p\left(t|\hat{\theta}_d\right) + \alpha \\ &\propto -\sum_{t \in q} p\left(t|\hat{\theta}_q\right) \log_2 p\left(t|\hat{\theta}_d\right) \end{aligned} \tag{6.3}$$

where $p\left(t|\hat{\theta}_q\right) \log_2 p\left(t|\hat{\theta}_q\right)$ is a document-independent constant, simplified as α, which is essentially the entropy of the query q. The constant α can be dropped, because it is uniform to all documents, and does not affect ranking of the documents.

The flexibility of the KLD-based model is that it allows us to model documents and queries in different ways. For a document, a straightforward way to build a language model

Table 6.2: Summary of notations in KLD framework in AS. Both document and query representations are extracted based on the style markers selected for indexing.

notation	meaning
t	Tokens that are extracted for indexing
d	The document representation after extraction of style markers
q	The query representation after extraction of style markers
C	The collection representation after extraction of style markers
$\hat{\theta}_q$	A query model for generating a query representation q
$\hat{\theta}_d$	A document model for generating a document representation d
$\|d\|$	The length of the document representation
$\|q\|$	The length of the query representation
$f_{t,d}$	The frequency of a style marker t occurring in document representation d
$f_{t,q}$	The frequency of a style marker t occurring in query representation q
$f_{t,C}$	The frequency of a style marker t occurring in collection representation C
μ	The tuning parameter in Dirichlet smoothing
$KLD\left(\hat{\theta}_q \| \hat{\theta}_d\right)$	The divergence between the document d and query q

is to use maximum likelihood. However it is less straightforward to model queries in IR applications, as most of the queries are short—typically only a few words—it is not easy to approximate probability distributions for query terms. Therefore, in IR the main difficulty in using a KLD model for retrieval is the estimation of the query model [Lafferty and Zhai, 2001; Liu and Croft, 2002; Shen et al., 2005; Zhai and Lafferty, 2001b]. In contrast to IR, in AS queries are in fact individual documents or a group of documents. Therefore it is reasonable to build entropy models for both the queries and the documents in the same way, and the differences between query models and document models can be measured using Equation 6.3.

However if $p(t|\hat{\theta})$ is 0 for some t, the divergence is undefined. This issue is addressed by applying Dirichlet smoothing to estimate probabilities from a background model [Zhai and Lafferty, 2004].[2] After smoothing, the probability of a term t generated by a document

[2]Details are presented in Chapter 4.

model $\hat{\theta}_d$ can be estimated by:

$$\begin{aligned} p'(t|\hat{\theta}_d) &= \frac{f_{t,d}}{\mu + |d|} + \frac{\mu}{\mu + |d|} p\left(t|\hat{\theta}_C\right) \quad \text{where} \\ |d| &= \sum_{t \in d} f_{t,d} \quad \text{and} \\ p\left(t|\hat{\theta}_C\right) &= \frac{f_{t,C}}{|C|} \end{aligned} \tag{6.4}$$

where $p\left(t|\hat{\theta}_C\right)$ is the probability of the token or style marker t in the background model $\hat{\theta}_C$; as we presented in Chapter 4, such a background model provides generalised statistics on the tokens, which is normally derived from large data sets. The parameter μ controls the mixture of the document model and the background model. The background probabilities dominate for short documents, as the evidence for the in-document probabilities is weak; the influence of the background model is less significant in longer documents. Following the discussion in Chapter 4, we believe that style markers $t \in q \cup d$ should be considered for KLD computation in AS rather than $t \in q$, as used in IR similarity measurement.

Combining Equation 6.3 with Equation 6.4, the divergence formulation can be expressed as:

$$KLD\left(\hat{\theta}_q||\hat{\theta}_d\right) \propto -\sum_{t \in q \cup d} \left(\frac{f_{t,q}}{\mu + |q|} + \frac{\mu}{\mu + |q|} p(t|\theta_C)\right) \log_2 \frac{f_{t,d}}{\mu + |d|} + \frac{\mu}{\mu + |d|} p\left(t|\theta_C\right) \tag{6.5}$$

For each query given to the AS system, divergence is calculated for each document-query pair throughout the collection. The documents whose entropy has the lowest divergence from the entropy of a query are, we propose, the most likely to share a similarity of writing style and thus should be returned the highest rankings. That is, the ranking is from the smallest KLD value to the highest. A basic exhaustive KLD-ranking algorithm for AS is shown in Algorithm 1; a shortcoming of such an approach is that, every single document in the collection is computed for every single query provided to the system, but only a tiny proportion of documents are returned, meaning that efficiency is an issue, however the computational complexity is almost linear. In the following sections, we design a series of experiments to examine the proposed AS approach. We believe that this data presents a difficult challenge for AA or AS, as, compared to novelists or poets, journalists do not necessarily have a strong authorial style, and the work may have been edited to make it consistent with a publication standard.

Algorithm 1 *KLD exhaustive ranking algorithm to identify the top n documents that are most likely to share the same authorship as that of a given query.*

$T \leftarrow$ Define style markers
Given T, generate $\hat{\theta}_C$ for smoothing
Given T, extract q
Calculate $p\left(t|\hat{\theta}_q\right)$ for each style marker in q
for $d \in C$ **do**
    Given T, extract d
    Set $KLD \leftarrow 0$
    for $t \in q \cup d$ **do**
        Calculate smoothed $p'\left(t|\hat{\theta}_q\right)$, using $\hat{\theta}_C$
        Calculate smoothed $p'\left(t|\hat{\theta}_d\right)$, using $\hat{\theta}_C$
        Set $KLD \leftarrow KLD - p'\left(t|\hat{\theta}_q\right) \log_2 p'\left(t|\hat{\theta}_d\right)$
    end for
end for
Identify the n smallest KLD

6.5 Experiments: Authorship Search

As data, we use the newswire collections that were introduced in Chapter 3—*AP10k*, *AP100k*, and *AP500k*. The collections consist of 10,700, 100,700, and 500,700 documents respectively. The *AP10k* and *AP100k* collections are drawn from the *AP* sub-collection of TREC data [Harman, 1995], while *AP500k* data contains documents from not only *AP*, but also *WSJ* and *SJM* (these two collections also consist of newswire articles). The seven candidates—the same as selected in *AP7* data—are used as the target authors to keep the consistency with the other AA investigations in this thesis. 100 documents authored by each of the seven authors are randomly selected (700 documents in total); all three newswire collections contain these 700 documents. In an ideal AS scenario, given a query—that is, a document or a set of documents—as the input, the top 100 returned documents should be authored by a certain author, who should also author the query document. In other words, each of the seven authors' writing styles have a total of 100 documents in the collection that are considered to

be "relevant" to a query for that style.

To define style statistically, all queries and documents are pre-processed and represented by sequences of style markers; test collections are also indexed with pre-defined style markers. The background models are important in such approaches; each is derived from the entire AP collection of over 250,000 documents in accordance to the particular type of style marker. An alternative is to use the collection as the background model in each case, but we decide to hold the background model constant across all experiments, which is from a much larger text data repository.

We evaluate our proposed AS system from several perspectives. The scalability of effective search is first examined by experimenting with collections of different size; then by reducing the volume of queries. Different indexing strategies are compared, to explore which is the most effective. The proposed KLD-based similarity model and other retrieval techniques are tested and compared. Finally we explore use of AS as an attribution method.

6.5.1 Feasibility and Scale in Size

In our first experiment we examine whether AS is feasible for small and large collections, using the proposed KLD similarity measure based on entropy. We have suggested in Chapter 4 that, a predictive profile of an author's writing style requires a sufficient number of training documents. It is observed that, approximately 400–500 training documents have been consistently effective for AA in our previous experiments. Therefore, the first seven queries used in this experiment are generated by concatenating 500 randomly selected documents written by each of the seven authors; this is to make sure the query does have identifiable writing style. Documents used for query constitution are distinct from the 100 in-collection documents. We refer these as the *500-document queries*; the style markers are function words in this experiment. The next seven queries are formed by concatenating the 100 in-collection documents; we call these the *100-included queries*.

Ideally, the returned documents by AS should have identical authorship. In our first experiment, we look at the top-100 ranked list as returned by the AS for each query. Table 6.3 presents the results on *AP10k* collection, as a confusion matrix. As shown, with the 500-document queries, those of author Currier and Dishneau are the most effective, while the

Table 6.3: The number of correct matches in the top 100 documents in response to each query, on the AP10k collection, using 500-document queries.

Query	Correct # of retrieved documents						
	Schweid	Currier	Skidmore	Dishneau	Kendall	Crutsinger	Beamish
Schweid	**48**	0	1	0	0	3	14
Currier	0	**61**	0	0	0	0	0
Skidmore	0	4	**35**	0	1	20	1
Dishneau	0	0	0	**61**	0	0	0
Kendall	0	1	3	0	**44**	2	0
Crutsinger	0	4	11	0	2	**52**	1
Beamish	14	0	0	0	0	1	**30**

Table 6.4: The number of correct matches in the top 100 documents in response to each query, on the AP10k collection, using 100-included document queries.

Query	Correct # of retrieved documents						
	Schweid	Currier	Skidmore	Dishneau	Kendall	Crutsinger	Beamish
Schweid	**59**	0	1	0	0	3	10
Currier	0	**58**	0	0	0	0	0
Skidmore	0	0	**49**	0	2	24	1
Dishneau	0	0	0	**61**	0	0	0
Kendall	0	0	1	0	**46**	0	0
Crutsinger	0	11	8	0	2	**56**	1
Beamish	11	0	0	0	0	1	**37**

query of Beamish is the worst. We next evaluate the 100-included queries on the same collection. The results, as shown in Table 6.4, are slightly better than using the 500-document queries in most cases; this can be attributed to the fact that the query documents are included in the collection, however the difference is small. Although these queries are formed from a smaller number of documents, as we can observe that they are highly consistent with the 500-document queries. It suggests that some authors do have distinct writing style, and that style can be effectively identified for some authors.

We then evaluate the seven 500-document queries on the other two larger collections, AP100k and AP500k, to examine the scalability of our model. Results are averaged from all query evaluations, and presented by plotting precision against recall, as shown in Figure 6.2.

We achieve 84.2% as the average precision at 10 documents retrieved, on the AP10k collection, 74.2% on the AP100k collection, and 30.0% on the AP500k collection of over half a million documents. As the density of correct matches in the collection falls from 1% to 0.02%, the effectiveness of query evaluation drops. Achievement of high recall is much more difficult with the largest collection, but the results show that AS is indeed feasible on even half a million documents, given appropriately constructed queries.

Another dimension of scale is the volume of training data available. In the experiments above we had a large volume of text of a particular author to constitute each query. With less text, effectiveness may decline. In the next experiment, for each author we construct 5 100-document queries and 25 20-document queries, by further splitting the 500-document queries randomly. Each of them is evaluated on AP10k collection; average results are shown in Figure 6.3. It can be seen that reducing the amount of training data does indeed reduce effectiveness of retrieval. For low levels of recall, both 100-document queries and 100-included queries result in reasonable effectiveness. Surprisingly, whether query documents are included in the collection does not have strong effect on the search results. This indicates to some extent that style markers such as function words are moderately stable within the documents of a particular author. However, queries of 20 documents are much less effective. While reasonable numbers of correct documents are still found in the top 10–30 answers, subsequent results are worse.

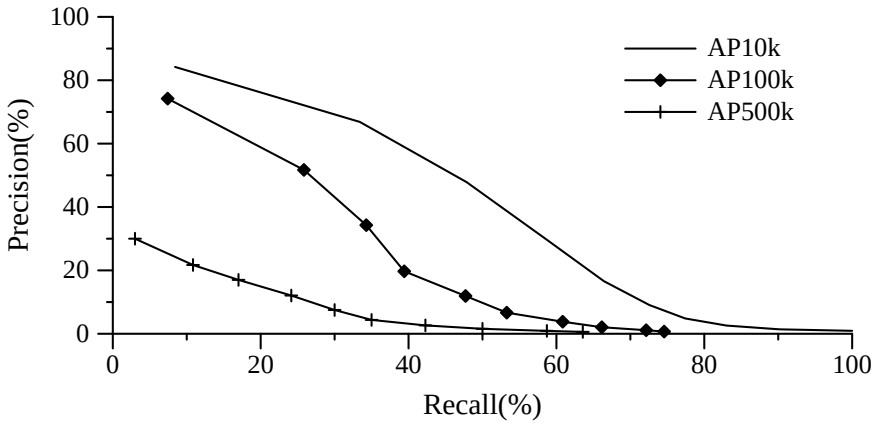


Figure 6.2: Precision versus recall for 500-document queries on each of the three collections: AP10k, AP100k, and AP500k.

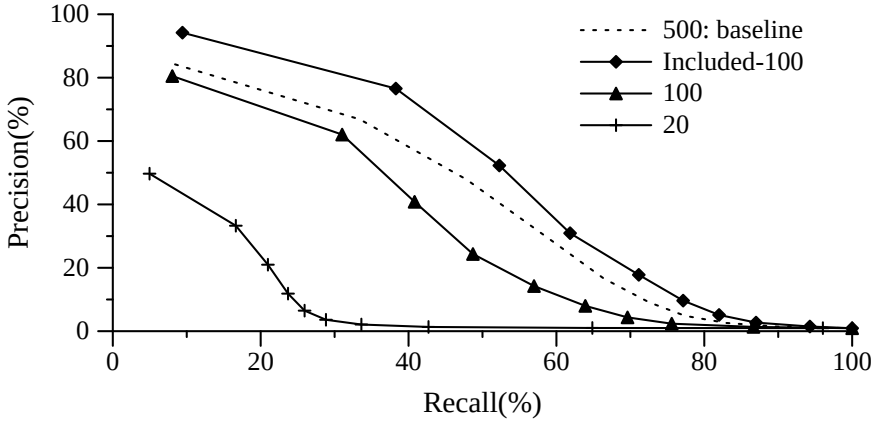


Figure 6.3: Effectiveness for queries composed of 20–500 documents, on the AP10k collection.

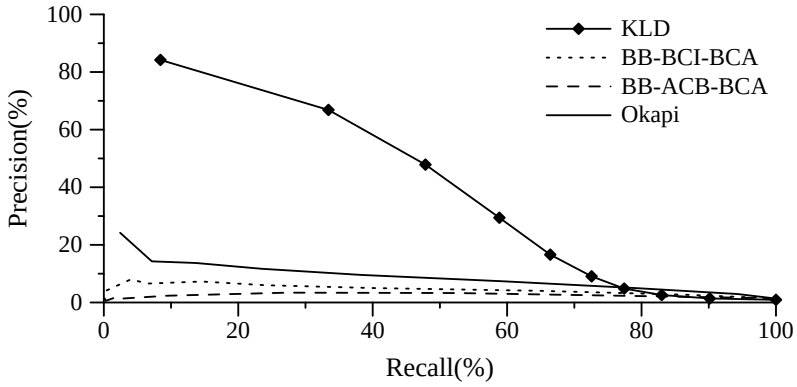


Figure 6.4: Effectiveness of different similarity measures on AP10k collection, using the 500-document queries.

6.5.2 KLD Ranking versus Other Measures

We have shown the feasibility of using KLD ranking methodology for AS in the previous experiment; it is worth exploring whether other successful IR similarity measures are plausible alternatives. In addition to the KLD ranking method, we use three measures that have been successfully used in IR, including OkapiBM25 and the vector-space measures BB-BCI-BCA and BB-ACB-BCA [Zobel and Moffat, 1998; 2006].

We use the same index and smoothing methods for all ranking schemes; evaluations are carried out with the seven 500-document queries, providing sufficient training samples, on the AP10k data. Results are averaged from 7 authors; precision-recall curves are plotted in Figure 6.4.

The IR similarity measures are surprisingly poor—none proved suitable for AS. The OkapiBM25 measure is slightly better than the other vector space models but based on the results, none is usable. We also empirically adjusted the tunable parameters in different models, achieving no significant improvement. Queries with a smaller volume of text intuitively lead to worse effectiveness. The reason why these measures are ineffective for AS is unclear and needs further investigation.

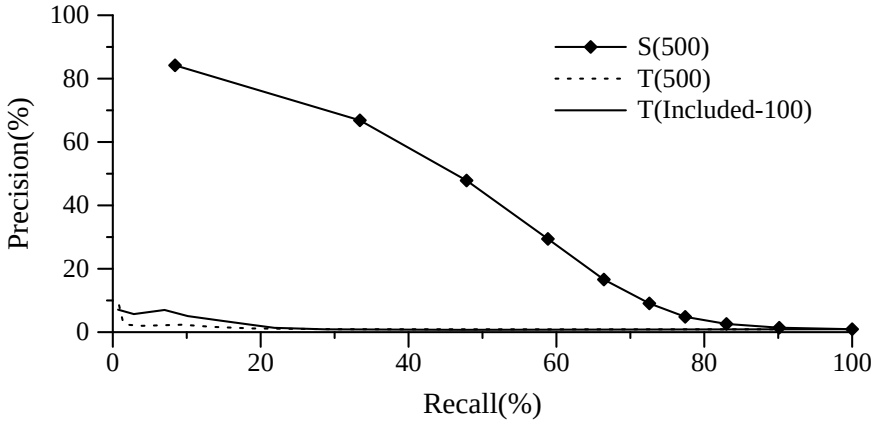


Figure 6.5: Comparison of using different indexing methods: function words vs. topic words on AP10k collection.

6.5.3 Index with Different Style Markers

In text categorization, documents are usually indexed or represented by topic words (bags-of-words) that occur in the documents [Bekkerman et al., 2003; Lai and Wu, 2002; Lewis et al., 2004; Yang, 2001]. However, in AA whether topic words are appropriate style markers is controversial; some researchers have used them, but most have not. In this experiment we contrast use of function words and topic words for AS based on the KLD similarity measure, using the AP10k collection. Results are shown in Figure 6.5 as precision-recall curves.

In Figure 6.5, the uppermost curve uses the 500-document queries and is the same as that of Figure 6.2; the dashed line is the comparable curve using topic words as style markers; and the solid line is based on topic words with the 100-included queries. As can be seen, AS with topic words completely fails; the results are slightly better than random. The results suggest that topic words are misleading for characterizing author writing style in a large document collection.

Other kinds of style markers are more plausible. For the next experiment, we use a list of 183 POS tags, 363 function words, and combination of both kinds of features to index collections. The 500-document queries are evaluated on all three collections of different sizes. Results are shown in Figures 6.6, 6.7, and 6.8. In the figures, "S" refers to function words, "POS" is part-of-speech, and "S-POS" indicates the combination of both marker types.

Function words consistently lead to greater effectiveness than POS tags with all three col-

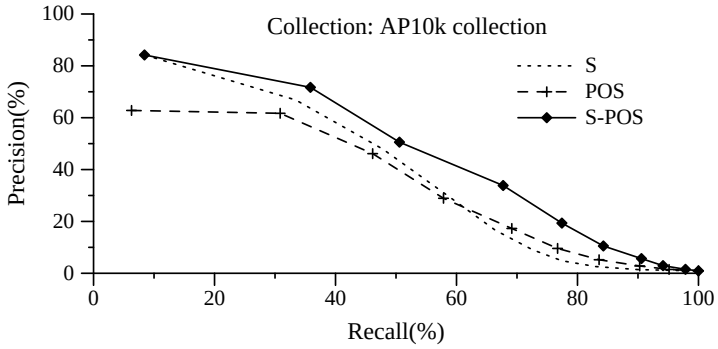


Figure 6.6: Effectiveness of different style markers on the AP10k collections, using the 500-document queries.

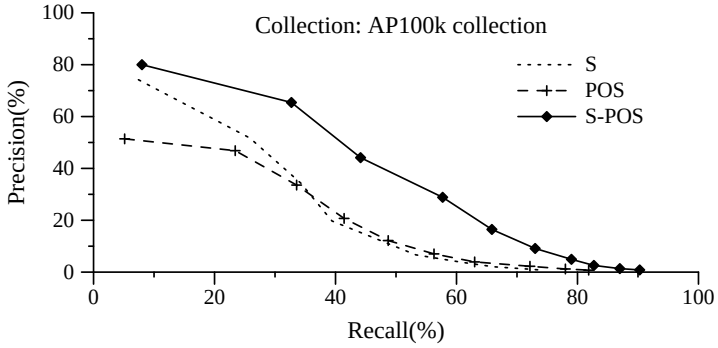


Figure 6.7: Effectiveness of different style markers on the AP100k collections, using the 500-document queries.

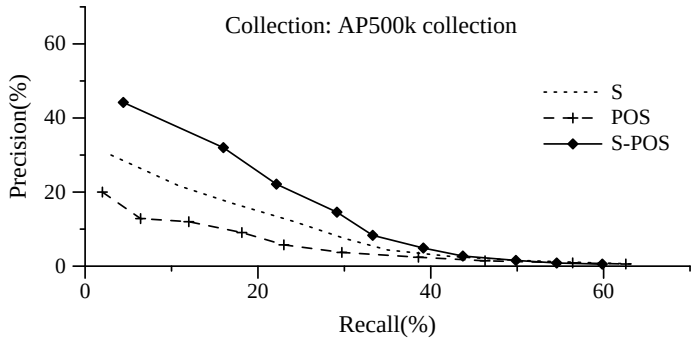


Figure 6.8: Effectiveness of different style markers on the AP500k collections, using the 500-document queries.

lections, which is consistent with the previous work presented in Chapter 4 and Chapter 5. However, indexing by the combination of function words and POS tags leads to even better effectiveness. With the AP10k collection, function words are almost as good as the combined features, and both approaches achieve the same precision at 10 documents retrieved of 84.2%. However, with increased collection size the advantage of combination increases. In particular on the AP500k collection, function words only achieve 30.0% precision at 10 documents retrieved, while addition of POS tags increases this precision to 44.2%. Although, the numerical differences are not small, it is worth conducting a significance test on these results. We used a paired t-test, using the combined results produced by AP10k, AP100k, and AP500k all together; we used p10 in our test. The results show that, at a confidence level of 0.01, POS works significantly worse than the other two marker types (with p-value of 0.0017 against the combined style markers, and 0.0092 against the function words). While the difference between using function words and combined features is smaller, which is significant at 0.05 level, but not 0.01 (with the p-value of 0.045). between function words These results show that, even though POS tags by themselves do not yield satisfactory effectiveness, they are helpful as an additional evidence of style, in particular for large data sets. Therefore we suggest that larger collections require more authorial evidence for effective AS.

We have previously observed that queries of some authors are easier to evaluate than others; it is therefore worthy studying the retrieval effectiveness on an author-by-author basis. In the next experiment, we explore how difficult it is to identify style for a particular author, by comparing various index schemes and by reducing the volume of query text.

Style Makes a Difference. From the reader's level of understanding, some author's style is easier to perceive than others. In a style-based AS system, queries having stronger writing style should lead to greater effectiveness than those of weaker style. Revisiting the results presented in Table 6.3, the query of Dishneau has retrieved 61 correct texts in the top 100 ranked documents, while Beamish has only 30 correct matches. We assume that Dishneau has a more distinctive style than Beamish, and the comparison between these two authors are demonstrated in Table 6.5, by changing the volume of the query text and types of style markers.

Table 6.5: Comparison between author Dishneau and author Beamish. Queries of different volume are evaluated on the AP10k collection. (As style markers, the notation "S" refers to function words; "POS" is part-of speech tags.)

		Query (S)				Other Markers	
Author	Precision	Inc(100)	Q(500)	Q(100)	Q(20)	POS	S+POS
Dishneau							
	p@10	100	100	100	100	100	100
	p@50	100	100	100	47	100	86
	p@100	61	61	61	58	67	52
Beamish							
	p@10	90	60	56	10	50	70
	p@50	56	40	34	4	54	44
	p@100	37	30	22	3	34	31

Queries of Dishneau perform consistently superior to those of Beamish; evaluation of queries formed by small volumes of text, as shown in the table, is dramatically different for Dishneau and Beamish. The queries of Dishneau lead to consistent success, while the queries of Beamish are more sensitive—the effectiveness varies significantly—and the accuracy is always worse than queries of Dishneau. With 20-document queries, the p@10 of Beamish dramatically degrades from 90% to only 10%; however the query of Dishneau leads to perfect 100% precision at top 10 retrieved documents. Similar degradation is observed with part-of-speech (POS) tags, as well as the combination of both POS and function words. When increasing the collection size to half a million documents, the query of Dishneau retrieves 32 correct matches; for Beamish, only 3 documents are correctly retrieved in the top 100 ranked documents.

The observation suggests that some authors' writing habits are easier to define or represent than others; the failure of AS on some cases may be caused by the lack of writing style of a certain author, more than by the method itself; we explore this further in the next

Algorithm 2 *The KLD Search-based AA algorithm to identify authorship for a given query in large document collections.*

$T \leftarrow$ pre-defined style markers
Set $t \leftarrow$ threshold value
Given T, generate q
Given q: Run **Algorithm 1**
Return a list of top n documents with highest ranked KLD, $L = \{d_j | j = 1, \cdots, n\}$
Extract k distinct authors, $\{a_i | i = 1, \cdots, k\}$, from L
Create an array of k elements: $\{v_i | i = 1, \cdots, k\}$, and each v_i is associated with an a_i.
for $j = 1$ to n **do**
    identify authorship for d_j
    Increment v_i by 1 for the identified a_i
end for
if $i = argmax_i \, \Phi(v_i)$ **then**
    Set $A \leftarrow a_i$
    if $\Phi(A) \geq t$ **then**
        Return A
    else
        Return "Unattributed"
    end if
end if

chapter.

6.5.4 Applicability to Authorship Attribution

The feasibility of effective AS motivates us to explore whether AS can be used for AA. Given such large collections, and queries with unknown authorship, an AS system may be able to assign an author to each query, by analysing the top ranked results from each search. We propose a search-based AA algorithm, shown in Algorithm 2.

We have a query for which authorship is unknown, refers to as "A". Using AS, a list

of n top-ranked documents is returned; these documents are written by k distinct authors. For each author a_i, a count v_i is calculated from the ranked list, that is, the number of documents by a_i. As shown in Algorithm 2, a function Φ of v_i is defined. The algorithm is more a framework, under which any function leading to an output within the range of $[0, 1]$ would be fine; t is a threshold between 0 and 1. A simple way to attribute authorship is to select the A with the largest v_i. In this case, the function Φ is defined as:

$$\begin{aligned} \Phi(v_i) &= \frac{v_i}{n} \\ n &= \sum_{i=1}^{k} v_i \end{aligned}$$

More strictly, a threshold t can be selected so that the query can be assigned to a particular author A if:

$$\begin{aligned} A \leftarrow i \quad \text{if} \quad i &= argmax_i(v_i) \quad \text{and} \\ \Phi(v_i) &\geq t \end{aligned}$$

Increasing t should reduce the likelihood of incorrect attribution. Richer functions Φ can be derived by taking the rank of a retrieved document into account, as well as the weight of the divergence computed. We are more interested in the feasibility of using such a framework for AA on large document collections; complicating Φ is of weak importance without such investigation.

To test these methods we use the *APvote10k* and *APvote100k* collections (see Chapter 3). The APvote10k collection includes 10,000 documents from 342 authors, and the APvote100k collection consists of 100,000 documents by 2,229 authors. In both collections, 100 documents of each of the seven test authors are included, which are distinct from query documents. The number of texts by each author varies significantly, from 1 to 835. In both collections more than 10% of the distinct authors have written over 100 documents each. All documents in 10k-vote have identified authorship, while in the 100k-vote collection more than 90% of the texts have identified authorship. As style markers we use the combination of function words and POS tags.

Results from previous experiments show that it is feasible to search for documents written by the same author as that of the query, given a group of documents of known authorship as

Table 6.6: The attribution results obtained using AS. There were 700 1-document queries and 70 10-document queries. The results are the number of queries (1-document and 10-document queries) that are correctly attributed, on the APvote10k collection, in the top 10, 20, 40, 60, 80, and 100 answers retrieved. The collection used is APvote10k

		Number of answers retrieved					
Queries	N_q	10	20	40	60	80	100
1-doc	700	357	343	334	346	335	337
10-doc	70	52	55	58	56	55	56

the query. In this experiment the authorship of the query is unknown and is to be identified.

As queries, 500-document queries are unreasonably large, therefore, we construct 100 queries of each author. Each query is a single document; these are referred to as 1-document queries. We also concatenate 10 randomly selected documents from the 100 document pool as a query; each author has 10 queries, named as 10-document queries. We experimented with these queries that are formed from individual documents and from 10-document sets; none of the query documents are in the collections.

Results are shown in the Table 6.6, using the threshold $t = 0$ so that attribution is made to the authorship of the biggest v_i. Evaluation is based on the top n ranked documents, for n from 10 to 100. As can be seen, queries can be effectively attributed using the APvote10k collection using only the top 10 documents retrieved; with both 1-document and 10-document queries, increasing n does not help.

With the 700 1-document queries, the overall correctness of attribution is 51.0%. Previous methods achieve this accuracy only on small collections. Greater attribution effectiveness is achieved with 10-document queries, giving overall 74.3% correct attribution. There has been no previous attempt at multi-class AA with more than a few authors. Both the number of authors and the size of the collection used are much more substantial than in all previous AA work; our results are a dramatic improvement on previous work.

We have observed strong inconsistencies amongst queries based on the works of differ-

Table 6.7: Voting-based AA results for each author; for each author there are 100 1-document queries and 10 10-document queries on APvote10k. N_q^c is the number of queries that are correctly attributed. $\overline{N_r}$ refers to the confidence value, that is the average number of documents in the top-k list that have the identical authorship as the input query. The collection used is APvote10k.

	1-document queries		10-document queries	
Author	$N_q^c/100$	$\overline{N_r}/10$	$N_q^c/10$	$\overline{N_r}/10$
Schweid	39	3.2	8	3.6
Currier	69	9.2	10	8.0
Skidmore	36	4.4	1	2.0
Dishneau	76	9.8	10	10.0
Kendall	58	4.8	10	7.4
Crutsinger	54	5.5	10	6.3
Beamish	25	2.7	3	3.0

ent authors, which are consistent with the observations in the previous chapters. Results extracted from top-10 lists are shown in Tables 6.7 and 6.8. As can be observed, queries using documents by Currier and Dishneau are more effective than other queries, not only in accuracy of AA, but also in confidence. This observation is consistent with results from previous search experiments.

The confidence is indicated by the average number of correct documents in the top-k ranked list, annotated as $\overline{N_r}$ in both tables. For instance, on the 10k-vote collection, the 100 1-document queries of Dishneau can be correctly attributed at 76% accuracy, providing $\overline{N_r} = 9.8$. Note that, unsurprisingly, the effectiveness of attribution for the 10-document queries is generally better than for the 1-document queries.

Another interesting observation is that, after evaluating the 700 1-document queries, the number of "unattributed" queries is greater than the number of "wrongly-attributed" queries, by approximately 10%. Almost 30% of the queries remain unattributed; this is due to the fact that there is more than one author that has the same value of v_i, so that the simple method of selecting the largest v_i is not capable of making a clear decision about

Table 6.8: Voting-based AA results for each author; for each author there are 20 1-document queries and 5 10-document queries on the APvote100k collection; for some authors only negligible numbers of correct documents were found; these are shown as negl. *(if less than 20% of queries are attributed).* N_q^c *is the number of queries that are correctly attributed.* $\overline{N_r}$ *refers to the confidence value, that is the average number of documents in the top-k list that have the identical authorship as the input query. The collection used is APvote100k.*

	1-document queries		10-document queries	
Author	$N_q^c/20$	$\overline{N_r}/10$	$N_q^c/5$	$\overline{N_r}/10$
Schweid	*negl.*	*negl.*	*negl.*	*negl.*
Currier	14	4.8	3	7.0
Skidmore	*negl.*	*negl.*	*negl.*	*negl.*
Dishneau	15	5.2	5	7.4
Kendall	8	2.9	5	4.4
Crutsinger	*negl.*	*negl.*	*negl.*	*negl.*
Beamish	*negl.*	*negl.*	*negl.*	*negl.*

authorship. As mentioned before, by changing the function Φ, we can make improvements to the basic algorithm. We test a refinement where, if the query cannot be attributed by Algorithm 2, then we consider the rank of each result; the idea is similar to MAP in IR. Using this refinement, a further 10% improvement in overall effectiveness is achieved. This result demonstrates that it is highly feasible to attribute even a short document in a large text collection that consists of a few hundred authors.

We also tested the proposed method on the 100k-vote collection (results are reported in Table 6.8), which has over 2,000 known authors. This experiment is less successful, with near-zero accuracy in four of the seven cases. Interestingly, these failures correspond to the results of lower confidence on the 10k-vote collection. For queries based on documents by Currier and Dishneau, the attribution accuracies are respectively 70% and 75%, suggesting 48% and 52% confidence. Again, use of 10-document queries leads to greater effectiveness and confidence. However, it can be seen that AA on larger collections, with a larger numbers of authors remains a challenge.

6.6 Chapter Summary

In this chapter we have proposed the novel task of authorship search (AS). AS aims to find documents written by a particular author, given appropriate queries. Unlike queries in an IR system, which are formed by a few words, queries in AS are formed from a group of training documents. Our proposal was that simple entropy-based similarity measure, and characterization of documents by distributions of style markers, can be successfully used for effective and scalable AS. The results have shown that such a method can be highly successful for collections of moderate size.

The proposed similarity measure, the Kullback-Leibler divergence, which is used to compute relative entropy, is far more effective than standard measures drawn from information retrieval, including vector space models and probabilistic Okapi models. From the indexing point of view, the conventional bag-of-words based strategy has clearly failed for style search, whereas both function words and POS tags, that is, the style markers, have shown to be reasonably effective. Function words have consistently led to greater effectiveness than that of POS tags; however for larger collections, combination of both kinds of markers have achieved even better results. On a collection of even half a million documents, 44.2% precision at top 10 retrieved documents has been achieved. To the best of our knowledge, this AS system is the first system that is able to search for documents based on authorship rather than subjects or topics. The success of the method is highlighted by the fact that we have used experimental data, newswise articles, that we regard as challenging for this task: in contrast to material drawn from sources such as literature. For such data, we would not expect human readers to be aware of strong stylistic differences between the authors.

Another major contribution of this research is that our AS system can be further used for authorship attribution. Previous methods struggle to correctly attribute authorship when given more than a few hundred documents or more than a few authors. Our method based on AS has achieved reasonable accuracy with 10,000 documents from several hundred authors. While it did not successfully scale to the collection of 100,000 documents in our experiments, this approach is more effective and more scalable than previous methods, and is a clear demonstration that authorship attribution can be applied on realistic collections.

To further demonstrate the robustness of our methods, in the next chapter, we evalu-

ate both classification-based and search-based AA approaches to a different domain from newswire—that is, another testbed drawn from English literature.

Chapter 7

Authorship Attribution in Classic Literature

In previous chapters, we have investigated both authorship attribution (AA) and authorship search (AS) primarily using collections of newswire articles. Although these documents are considered to be more challenging in contrast to literature, as discussed in Chapter 4, it is worth evaluating both of our proposed approaches on a collection of English literature, where the study of writing style originated. In this chapter we explore whether the works of authors of classic literature can be correctly identified with either of the entropy-based attribution model, or the entropy-based search model, and to understand when and why attribution fails in some cases.*

7.1 Background

It is well accepted that certain authors have a highly distinguishing writing style, in particular renowned novelists. Readers can recognize works of their favorite writers with little difficulty. However style is not easy to define or identify automatically. The concept of style underlies the AA approaches that have been effectively applied to collections of newswire stories in our previous investigations.

As noted in Chapter 2, in most AA work the style markers have been distributions of

*This chapter incorporates work originally published by Zhao and Zobel [2007a].

elements that can be extracted from texts, such as function words [Burrows, 1987; Binongo, 2003; Baayen et al., 2002; Juola and Baayen, 2003; Holmes et al., 2001] and part-of-speech tags [Kukushkina et al., 2001; Stamatatos et al., 1999; 2001; Baayen et al., 1996]. Given such markers, an classification method is then required to identify a likely author from amongst a set of potential authors.

Using newswire data in previous chapters, we have explored several AA methods, including state-of-art machine learning approaches, finding that the best results were yielded by our statistical methods based on language models and entropy (details in Chapter 4). We also proposed and evaluated KLD-based authorship search (AS) in Chapter 6, which was further explored as a search-based method of AA. Different from IR, in AS, language models are used to match documents by style rather by topic.

Various types of markers, such as function words and POS tags, have been applied to the evaluation of both AA and AS. We found in Chapters 5 and 6, in agreement with other researchers, that function words are generally more reliable than other marker types for AA. The results achieved have shown to be more effective, and substantially more scalable than any method published in prior AA research. However, even though these methods are successful on average, they are not successful for all. Why this occurs has been unclear.

The aims of this chapter are to compare the effectiveness of search and classification as attribution methods; to see how effective attribution is on literature; and to understand when and why attribution fails in some cases. We use the proposed KLD-based principle; and establish a data set and the indexing methods to extract style markers. Language-model-based AA and AS approaches are applied to a corpus of novels extracted from *Project Gutenberg*.[1] While not a large corpus by text collection standards, it is more substantial than the collections used in most previous work in the area of AA. Our Gutenberg-based collection contains a substantial cross-section of 19th-century English literature as well as other work.

[1] See www.gutenberg.org.

7.2 Experiments and Results

It is intuitive that the task of AA may be relatively easy with a small collection if there are only a few authors, or if the authors are from widely different periods. In this respect, we seek to group literature that is representative and consistent. Authors are selected from the list of the top 100 most downloaded authors in the *Project Gutenberg*, on the date that we started collecting the data. In total 55 of the top 100 most downloaded authors are chosen (including playwrights as discussed below), and the total number of books collected is 634. We name this corpus *Gutenberg634*. In order to revisit a well-known AA problem—that is, the Shakespeare authorship debates—in the *Gutenberg634* corpus we also include plays by several major playwrights of the late sixteenth and early seventeenth century: Marlowe, Jonson, Beaumont & Fletcher (who wrote together), and Shakespeare.

7.2.1 Testbed: Gutenberg634

Unlike the *GutenbergSmall* data, a fragment-based corpus used in Chapter 4, *Gutenberg634* is a collection of complete books. In most cases we collect 10 books per author, or fewer if there are less than 10 that are available. However in some cases we collect all works for that particular author, such as William Shakespeare. Some factors are taken into account in selecting the books. We avoid choices that we feel are not consistent with the aims of our experiments:

- Poetry, dictionaries, images, or text in languages other than English are not considered.
- Individual short stories are avoided, especially in cases where a collection containing the story is also available.
- Authors, who are in the top 100 most downloaded author list but with four or fewer books, are not considered. This is to assure that all selected authors have sufficient data.[2]
- We keep both plays and novels. However plays are greatly in the minority, which are written by renowned playwrights from the time of Shakespeare.

[2]The complete list of authors is shown in Table 7.3 and Table 7.4.

However maintenance of consistency is not straightforward. After filtering based on the above criteria, problems still exist:

- It is common that a book has many different editions; each of these multiple versions of the same book has a distinct entry for downloading; and, texts of these entries are identical, unless the version of that book is written in languages other than English. Repeated versions of books introduce duplicates to the collection, and can cause an overestimation of the proposed methods.

- A book may be presented in different forms: as a complete book, a series of chapters, a set of groups of chapters, or a number of fragments. Each of the units has a separate entry for downloading; texts descriptors as "Volume 1" or "Chapter 1" are usually present in the downloading entry of that unit as an indication that it is a fragment. The existence of these types of documents also results in the collection having duplicates.

- Some available document entries associated with a certain author are not appropriate for style analysis. These materials are usually indicated by text included in the hyper-linked titles of the document entries, such as "as Translator", "as Illustrator", "as Editor", or "as Contributor". The semantic meaning of these descriptors tell us that the so-called author of these downloadable documents is not the actual author. In this case, these documents should also be avoided . However the challenge is that this kind of information may not be directly observable in the downloading entries of some books, and thus a manual check on the corresponding description pages is required to assure quality of the corpus.

Table 7.1 is an illustration of the inconsistency of books available in *Project Gutenberg*. As shown, 42 works of Shakespeare are appropriate for style analysis, and are collected into the *Gutenberg634* collection, whereas the total number of available document entries is 221—five times larger than what we collect. Another typical author is Twain, who has 202 entries available in total, however only 14 texts are considered to be appropriate for our investigation. In most cases we ensure the major works of a particular author are included, excluding duplicates and fragments.

Table 7.1: An example of the strong inconsistency of documents that can be fetched from the Project Gutenberg. The statistics in this table were collected when we started collecting the data, and do not represent the current updates on the Project Gutenberg.

	Number of Documents				
Author	Collected	Total	Non-English	Duplicates	Fragments & Others
Shakespeare	**42**	221	56	96	27
Twain	**14**	202	2	41	145

Within each volume, the raw text usually contains other-author material that should be eliminated from style analysis; examples are the Gutenberg disclaimer, editors' commentaries, and sometimes an introduction or preface written by someone else. Due to the observation that this kind of material is dramatically inconsistent amongst books, we thus manually delete all other-author text from each book.

7.2.2 Indexing Mechanism

An indexing mechanism is responsible for extraction of style representations of documents. Using this *Gutenberg634* collection, we test different forms of marker types: function words and part-of-speech tags (POS). Additionally, in contrast to collections that have been used in the previous chapters, documents in the *Gutenberg634* are much longer, having over 80,000 words on average. Therefore, in addition to use the POS tags individually, we also test bi-gram POS tags. Each document is indexed with the three types of style markers individually. Table 7.2 gives an example of how usage of different type of style markers can vary between authors. In this example from the collection of *Gutenberg634* data, for even common style markers, the usage can be quite different. In the table, both Shakespeare and Marlowe are the best-known playwrights. For common function words, Shakespeare uses "a" more frequently than Marlowe but makes less use of "the" and "of". On the other hand, Shakespeare uses conjunction (notated as "CC" in the table) and adjectives ("JJ" in the table) more frequently than Marlowe.

The approaches explored in this chapter have been described in Chapters 4 and 6. In these methods language models form the basis of approximations of distributions in relation to

Table 7.2: Usage statistics for the commonly used style markers for two authors. Each number is, for that author, the percentage of function word occurrences that is the particular function word. Counts are averaged across all documents available for each author.

	Function words			POS tags		
	the	of	a	CC	IN	JJ
Shakespeare	7.6	4.8	4.1	3.8	5.9	2.8
Marlowe	9.5	6.2	3.2	3.2	6.4	2.4

different marker types; Dirichlet smoothing is consistently used, with the parameter μ being carefully tuned, the choice of a background model is an important factor in such techniques. Based on our experience with KLD-based techniques from previous study in Chapter 4, we believe that the 634 books in the collection are not sufficient for deriving a comprehensive background model. Therefore, the entire AP collection consisting of over 250,000 newswire articles is chosen as the background model. The AP data has been fully tagged with POS tags as part of our earlier experiments in Chapter 6. This tagged-AP corpus is used to derive a background model for the style markers of POS tags and POS tag pairs. In some respects this choice is not ideal, as it consists of non-fiction written in the late 1980s and early 1990s, but it is the best option available to us. As the results show, AA is highly effective with the background models drawn from AP and tagged-AP data; a better background model may further improve results, but they are already strong.

7.2.3 Classification-based Authorship Attribution

In the first experiment, we use the KLD-based classification method for AA, using different marker types. Evaluation is carried out using leave-one-out cross validation, that is, each of the 634 books is left out in turn to be identified. As we discussed in Chapters 2 and 3, the evaluation makes an one-class AA investigation. In total, we have 634 runs for each of the three kinds of marker. The aim is to make a decision on authorship for each of the left-out works: whether the text is by a given author or is more likely to be by someone else. This is technically one-class AA, as introduced in Chapter 2. Take the 42 Shakespeare's works for example; a total number of 42 decisions are made. For each run we create a positive

Shakespeare model using the remaining 41 of his texts, excluding the left-out work, and create a negative model using the 592 texts by the authors other than Shakespeare. The left-out book is then examined using the divergence with both positive and negative models for making the final attribution decision. This is repeated for each of Shakespeare's 42 texts and each of the 55 authors. Each book in the negative model is also left out in turn. Classification is on both positive and negative leave-one-out estimations. Positive classification accuracy, denoted as Acc_p in this chapter, can be measured for each author by:

$$Acc_p = \frac{Number\ of\ correctly\ attributed\ positive\ documents\ of\ an\ author}{Total\ number\ of\ positive\ documents\ of\ an\ author}$$

Negative classification results, denoted as Acc_n in this chapter, are measured by:

$$Acc_n = \frac{Number\ of\ correctly\ attributed\ negative\ documents\ against\ an\ author}{Total\ number\ of\ negative\ documents\ against\ an\ author}$$

Acc_p provides an estimate of the rate of false misses; and, Acc_n can be used to estimate the rate of false matches, for example, for a model trained on a particular author, say Austen, and a work by some other author, say Shakespeare, we intend to find out the likelihood that the work is misattributed as by Austen.

In these experiments, overall correctness of classification of negative examples, Acc_n, is over 95%, much higher than the accuracy on positive examples, Acc_papproximately 85%. We split the results of attributing positive samples into two tables in accordance with the effectiveness of using function words: the results shown in Table 7.3 are of 26 authors leading to more than 90% accuracy on function words; and Table 7.4 presents results of authors that are less effective. In these tables, *FW* refers to the style markers of function words, *POS* indicates the part-of-speech tags, and *PP* indicates the POS tag pairs. Some severe failures can be observed with the use of function words. Schiller and Tolstoy are problematic, as well as Wilde as shown in Table 7.4, achieving the positive accuracies of 70.0%, 53.5%, and 28.6% respectively. Another difficult author is Defoe, perhaps surprisingly, as he is the only author from the early eighteenth century. Overall, however, the results are highly satisfying.

The texts that we have used in these experiments are complete books, each having over 80,000 words on average. As such, it is worth exploring whether AA would be effective on smaller volume texts. To achieve this, we then re-run the positive leave-one-out experiments using the complete texts for modeling, while a single 1000-word fragment is extracted for

Table 7.3: Results (better than 90% on function words) of one-class authorship attribution. Results are total correct per author (N_c) and a percentage of correct attribution (Acc_p).

Author	FW		POS		PP	
(# of book)	N_c	Acc_p	N_c	Acc_p	N_c	Acc_p
Total(634)	543	85.6	527	83.1	528	83.3
Alcott(10)	9	90.0	9	90.0	8	80.0
Alger(10)	10	100.0	10	100.0	10	100.0
Austen(8)	8	100.0	7	87.5	7	87.5
Baum(10)	10	100.0	9	90.0	9	90.0
Churchill(22)	20	90.9	19	86.4	18	81.8
Collins(23)	21	91.3	18	78.3	19	82.6
Conrad(12)	12	100.0	11	91.7	11	91.7
Fletcher(6)	6	100.0	6	100.0	6	100.0
Hardy(7)	7	100.0	3	42.9	6	85.7
Henry(9)	9	100.0	9	100.0	9	100.0
Holmes(9)	9	100.0	9	100.0	8	88.9
Jonson(7)	7	100.0	7	100.0	7	100.0
Kingsley(10)	9	90.0	9	90.0	9	90.0
Kipling(8)	8	100.0	7	87.5	7	87.5
London(21)	21	100.0	21	100.0	20	95.2
Lytton(10)	9	90.0	10	100.0	10	100.0
Marlowe(5)	5	100.0	5	100.0	5	100.0
McCutcheon(10)	10	100.0	10	100.0	10	100.0
Motley(10)	10	100.0	10	100.0	10	100.0
Pepy(10)	10	100.0	10	100.0	10	100.0
Poe(6)	6	100.0	6	100.0	6	100.0
Rohmer(10)	10	100.0	10	100.0	10	100.0
Scott(10)	10	100.0	10	100.0	10	100.0
Shaw(10)	9	90.0	8	80.0	8	80.0
Shakespeare(42)	40	95.2	41	97.6	41	97.6
Stockton(10)	9	90.0	8	80.0	8	80.0
Twain(14)	13	92.9	12	85.7	12	85.7
Verne(10)	10	100.0	10	100.0	10	100.0
Wells(10)	9	90.0	8	80.0	8	80.0
Wodehouse(23)	22	95.7	20	87.0	21	91.3

Table 7.4: Results (less than 90% on function words) of one-class authorship attribution. Results are total correct per author (N_c) and a percentage of correct attribution (Acc_p).

Author	FW		POS		PP	
(# of book)	N_c	Acc_p	N_c	Acc_p	N_c	Acc_p
Burroughs(9)	8	88.9	8	88.9	8	88.9
Bierce(8)	6	75.0	6	75.0	5	62.5
Carroll(6)	3	50.0	3	50.0	2	33.3
Curtis(7)	6	85.7	5	71.4	5	71.4
Darwin(9)	6	66.7	6	66.7	7	77.8
Defoe(9)	5	55.6	5	55.6	5	55.6
Dickens(11)	8	72.7	6	54.5	6	54.5
Galsworthy(10)	8	80.0	5	50.0	5	50.0
Haggard(37)	26	70.3	31	83.8	30	81.1
Harte(9)	8	88.9	9	100.0	9	100.0
Hawthorne(10)	5	50.0	9	90.0	8	80.0
Howells(10)	6	60.0	6	60.0	6	60.0
James(19)	17	89.5	17	89.5	17	89.5
Lang(10)	7	70.0	2	20.0	4	40.0
Lever(9)	8	88.9	4	44.4	8	88.9
MacDonald(9)	7	77.8	5	55.6	7	77.8
Maupassant(9)	7	77.8	8	88.9	7	77.8
Parker(10)	8	80.0	10	100.0	8	80.0
Schiller(10)	7	70.0	9	90.0	9	90.0
Stevenson(10)	7	70.0	6	60.0	5	50.0
Tolstoy(15)	8	53.3	7	46.7	6	40.0
Wake(9)	6	66.7	9	100.0	9	100.0
Warner(10)	8	80.0	9	90.0	9	90.0
Wilde(7)	2	28.6	3	42.9	2	28.6
Yonge(10)	8	80.0	7	70.0	8	80.0

each text being identified; the book associated with the extracted fragment is excluded in modelling positive works. Each fragment is drawn from a few thousand words[3] after the start of the text. These experiments are not successful, achieving an overall accuracy of only 10.4%. Use of 10,000-word fragments is more effective, giving overall an accuracy of 53.2%. Even though the result is far from perfect, it is significantly better than random average accuracy of around 2%. At this level of accuracy, AA is not conclusive, but is nonetheless highly indicative.

We also observe that the effectiveness varies in accordance with different marker types. For 11 authors, all marker types produce perfect attribution results. POS tags are less effective in contrast to function words in general, however we do observe the POS tags are better markers in some cases. Perfect 100% attribution results are obtained for Harte, Parker, and Wake, given POS tags as style markers; while the results of using function words for these authors is less effective, in particular for author the Wake, giving only 67% positive accuracy. We had hoped that POS tags would prove the more powerful style markers, however they have been inferior compared to function words. Similar inconsistencies are also observed with POS tag pairs. The pairs should in principle give an indication of the way in which the author combines parts of speech, which is plausibly a signature of the author's writing style. However, automatic identification of POS may to some extent undermine this aim. Following the earlier discussion, we hypothesize that the very qualities that make an author's style unique may lead to tagging failures due to the lack of observations for training. That is, POS tags and POS tag pairs are likely to be least reliable for the most distinguishable characteristics in the text.

7.2.4 Authorship Search

In contrast to the classification-based experiment, we apply authorship search (AS) as a search-based attribution method to AA; details of the methodology are in Chapter 6. Attribution via search provides two functions: a way of identifying the author of a query document; and a way of finding other documents that the author has written within the collection. Our

[3]More specifically, we discard the first 1,000 lines of texts in each book; fragments are then extracted from the remaining texts.

aim in this experiment is to test the effectiveness of different style markers when used with search-based AA.

Each book in the collection is used as a query individually. The remaining books in the collection are then ranked according to their similarity with respect to the query book provided to the AS system. The hypothesis is that, if the collection is indexed with markers that are a good indication of style, then the documents ranked the highest should be by the same author as that of the query document.

Consistent with the previous experiments, three different forms of index schemes are used: function words, POS tags, and POS-tag pairs. We construct 634 queries for each form of the indexing schemes, and each is evaluated using the remaining 633 books from the *Gutenberg634* corpus, excluding that particular query document. We report the performance for each query measured with p@5, that is, the precision at the top 5 ranked results. Specifically, it is the number of works by the same author as that of the query in the top 5 retrieved results. Outcomes are split into Table 7.5 and Table 7.6 based on values of p@5 measured on function words. Authors with p@5 higher than 80% are grouped in Table 7.5, and the remaining authors are presented in Table 7.6.

In the tables, N_o indicates the optimal retrieval, that is, the maximum number of correct matches that can be retrieved for each author. For example, Alcott has 10 books that are used as the query in turn; ideally the optimal retrieval should be 50, five for each query book if only looking at the top 5 ranked results. Using function words, 43 out of 50 (or 86.0%) can be retrieved for Alcott; while results using other style markers are much lower. On average over 76% of the documents in the top 5 are correct matches, given function words for indexing. For 15 of the 55 authors the overall p@5 is 90% or better. Other markers are somewhat less successful, but still reasonably effective. With POS tags, an average of 62% p@5 is obtained; 66.5% for POS-tag pairs. The results of overall p@5 show that the KLD-based search with indexing on style markers can be an effective mechanism for matching texts by authorship, or style.

However, as in the previous experiment, search-based AA is not particularly successful for some authors. An elementary cause might be the number of training examples; the more documents of a particular author included in the collection, the easier the retrieval task is.

Table 7.5: Results of p@5 (greater than 80% on function words) of search-based attribution. Results are overall P@5 per author, a percentage of optimal retrieval (N_o). N_r is the number of correct matches in the top 5 retrieved books.

Author	FW		POS		PP	
(# of books) / N_o	N_r	p@5	N_r	p@5	N_r	p@5
Overall(634) /3165	2409	76.1	1962	62.0	2106	66.5
Alcott(10) / 50	43	86.0	32	64.0	32	64.0
Alger(10) / 50	50	100.0	47	94.0	50	100.0
Austen(8) / 40	39	97.5	31	77.5	37	92.5
Baum(10) / 50	45	90.0	42	84.0	44	88.0
Burroughs(9) / 45	39	86.7	21	46.7	27	60.0
Churchill(22) / 110	93	84.5	75	68.2	78	70.9
Collins(23) / 5	5	91.3	94	81.7	101	87.8
Conrad(12) / 60	51	85.0	24	40.0	32	53.3
Haggard(37) / 185	168	90.8	154	83.2	165	89.2
Hardy(7) / 35	35	100.0	18	51.4	15	42.9
Harte(9) / 45	36	80.0	36	80.0	37	82.2
Henry(9) / 45	40	88.9	37	82.2	40	8.9
James(19) / 95	80	84.2	48	50.5	44	46.3
Lever(9) / 45	40	88.9	40	88.9	38	84.4
London(21) / 105	101	96.2	64	61.0	79	75.2
Lytton(10) / 50	49	98.0	49	98.0	43	86.0
Maupassant(9) / 45	40	88.9	33	73.3	37	82.2
McCutcheon(10)/ 50	45	90.0	35	70.0	45	90.0
Motley(10) / 50	50	100.0	50	100.0	45	90.0
Parker(10) / 50	40	80.0	33	66.0	21	42.0
Pepy(10) / 50	50	100.0	50	100.0	50	100.0
Rohmer(10) / 50	50	100.0	46	92.0	48	96.0
Scott(10) / 50	50	100.0	49	98.0	50	100.0
Shakespeare(42)/210	203	96.7	197	93.8	199	94.8
Twain(14) / 70	57	81.4	33	47.1	46	65.7
Verne(10) / 50	41	82.0	35	70.0	46	92.0
Wodehouse(23) / 115	113	98.3	97	84.3	100	87.0

Table 7.6: Results of p@5 (less than 80% on function words) of search-based attribution. Results are overall P@5 per author, a percentage of optimal retrieval (N_o). N_r is the number of correct matches in the top 5 retrieved books.

Author	FW		POS		PP	
(# of books) / N_o	N_r	p@5	N_r	p@5	N_r	p@5
Bierce(8) / 40	6	15.0	4	10.0	5	12.5
Carroll(6) / 30	7	23.3	4	13.3	1	3.3
Curtis(7) / 35	19	54.3	9	25.7	12	34.3
Darwin(9) / 45	28	62.2	31	68.9	29	64.4
Defoe(9) / 45	22	48.9	20	44.4	19	42.2
Dickens(11) / 55	40	72.7	11	20.0	16	29.1
Fletcher(6) / 30	23	76.7	19	63.3	20	66.7
Galsworthy(10)/ 50	22	44.0	27	54.0	29	58.0
Hawthorne(10) / 50	30	60.0	31	62.0	32	64.0
Holmes(9) / 45	30	66.7	20	44.4	20	44.4
Howells(10) / 50	23	46.0	15	30.0	20	40.0
Jonson(7) / 35	19	54.3	26	74.3	30	85.7
Kingsley(10) / 50	28	56.0	17	34.0	13	26.0
Kipling(8) / 40	28	70.0	12	30.0	19	47.5
Lang(10) / 50	19	38.0	11	22.0	14	28.0
MacDonald(9) / 45	26	57.8	11	24.4	18	40.0
Marlowe(5) / 20	6	35.0	6	30.0	8	40.0
Poe(6) / 30	21	70.0	18	60.0	19	63.3
Schiller(10) / 50	19	38.0	21	42.0	22	44.0
Shaw(10) / 50	33	66.0	28	56.0	30	60.0
Stevenson(10) / 50	11	22.0	4	8.0	9	18.0
Stockton(10) / 50	38	76.0	23	46.0	33	66.0
Tolstoy(15) / 75	38	50.7	26	34.7	28	37.3
Wake(9) / 45	34	75.6	40	88.9	38	84.4
Warner(10) / 50	27	54.0	26	52.0	28	56.0
Wells(10) / 50	23	46.0	17	34.0	23	46.0
Wilde(7) / 35	2	5.7	2	5.7	1	2.9
Yonge(10) / 50	33	66.0	13	26.0	21	42.0

Results are somewhat better for authors with more texts available within the collection.

Alternative causes are attributable to style itself. In the collection there are four authors whose works are originally written in a language other than English. Schiller's works are in German, and Tolstoy's works are in Russian; the other two examples are Maupassant and Verne, both of whom originally wrote in French. As shown in Table 7.6, Schiller and Tolstoy are amongst the worst cases for AS indexed function words for example. The original identifiable style of the author may be removed by the process of translation, and multiple translators may be involved in translating a single book.

Interestingly, we observe a weak tendency for errors to be in the right period. For example, as discussed further below, when the works of Marlowe were used as queries, most of the retrieved matches are plays written by his contemporaries. However the errors and time period are not strongly correlated from the results. Finally, we suggest that not every author has a strongly identifiable writing style: some do have a weak style that is hard to identify; and others have inconsistent style, meaning that they may change their style between books. For example, considering the works of Wilde and Bierce, both of whom are satirists, it is somewhat not surprising that these works are difficult to attribute.

7.2.5 Shakespeare and His Contemporaries

We now revisit a famous AA problem, the *Shakespeare authorship debates* that we introduced in Chapter 1. In the *Gutenberg634* collection, we also include works by Shakespeare and his contemporaries for a case study, following the suggestion by some scholars that Shakespeare's plays were written by someone else.[4] As a preliminary investigation, we intend to know whether our methods could yield any evidence to support the authorship. The major playwrights of Shakespeare's time selected are: William Shakespeare, Francis Beaumont & John Fletcher (whose works are mostly co-authored), Ben Jonson, and Christopher Marlowe. By examining the extent to which these works are consistent with each other in terms of writing style, and whose works are likely to match to whose, we conjecture that it is possible to discover some evidence pointing in one direction or the other.

Our investigation is not flawless; an admitted weakness is that we have not been able to

[4]Some say that the proposition was first put by *Edward Blount* in 1623, others cite Queen Elizabeth I.

Table 7.7: Example ranked lists (top 5) for five works of Shakespeare; markers are function words only.

Rank	Sh.139	Sh.149	Sh.155	Sh.163	Sh.166
1	Sh.166	Sh.165	Sh.128	Sh.162	Sh.139
2	Sh.145	Sh.21	Sh.162	Sh.166	Sh.148
3	Sh.148	Sh.29	Sh.167	Sh.169	Sh.147
4	Sh.147	Sh.164	Sh.147	Sh.23	Sh.145
5	Sh.155	Sh.22	Sh.164	Sh.168	Sh.155

collect many candidates for such investigation. Unfortunately these works are not available to us in a suitable form. The texts being explored have been subjected to intensive literary analysis for several centuries by domain experts and scholars, and we do not claim that a straightforward statistical analysis, such as by our models, can lead to a new clear result. However it is assumed in AA that patterns of writing are not easily disguised or manipulated consciously so that we hope to observe inconsistencies in the statistical characters of Shakespeare's works, if the works are authored by someone else as supposed.

To examine this question of authorship, we further analyse the search results based on function words. We examine the ranked lists for selected books by each of these four authors: Shakespeare, Beaumont & Fletcher, Marlowe, and Jonson. For simplicity, we use a shorthand to notate the playwrights in the tables below in accordingly: "Sh.", "BF.", "Ma.", and "Jn.". Each shorthand is followed by a number that is an index assigned to a book name in the *Gutenberg634* corpus. For example Sh.165 represents the book of *Timon of Athens*. We provide results of evaluating five query books for each of these authors for discussion.

In Table 7.7, we list the top 5 retrieved authors and books for each of 5 example query books of Shakespeare. With the entire *Gutenberg634* corpus, when applied to the 42 Shakespeare works as queries, we find a high consistency of writing for Shakespeare, with overall 203 of 210 top 5 listings being correct. In Tables 7.8, 7.9, and 7.10 we show the top 5 ranking lists for Beaumont & Fletcher, Marlowe, and Jonson respectively. These results are rather less consistent than for Shakespeare. As earlier discussed, one of the possible causes may be the volume of texts available in the collection. There are far fewer training texts for these

Table 7.8: Example ranked lists (top 5) for works of Beaumont & Fletcher; markers are function words only.

Rank	BF.19	BF.20	BF.21	BF.22	BF.23
1	BF.24	BF.21	BF.20	BF.21	BF.20
2	BF.23	BF.23	BF.22	BF.20	BF.24
3	Sh.149	BF.22	BF.23	BF.23	BF.21
4	Sh.165	BF.24	BF.24	BF.24	BF.22
5	Sh.159	Jn.7	Jn.8	BF.19	Jn.8

Table 7.9: Example ranked lists (top 5) for works of Marlowe; markers are function words only. "other" indicates the returned author is not in the selected playwrights.

Rank	Ma.11	Ma.12	Ma.13	Ma.14	Ma.17
1	Sh.166	Ma.13	Ma.14	Ma.13	Sh.166
2	Sh.163	Sh.139	Sh.139	Sh.139	Sh.139
3	Sh.148	Ma.14	Ma.12	Ma.12	Sh.147
4	Sh.139	Ma.17	Sh.166	Sh.166	Sh.155
5	Sh.169	other	Sh.147	Sh.147	Sh.148

Table 7.10: Example ranked lists (top 5) for works of Jonson; markers are function words only.

Rank	Jn.1	Jn.5	Jn.7	Jn.8	Jn.9
1	Jn.8	Jn.7	Jn.5	Sh.142	Jn.8
2	Sh.162	Sh.168	Jn.2	Sh.167	Jn.2
3	Sh.28	Jn.1	Sh.168	Jn.2	Sh.147
4	Sh.142	Sh.8	Jn.8	Jn.7	Sh.139
5	Sh.155	Sh.156	Sh.167	Jn.1	Sh.26

authors than for Shakespeare. In the case of Beaumont & Fletcher, as shown in Table 7.8, only 6 of the 25 documents are mismatches. The cases of Marlowe and Jonson are more intriguing. As observed in Table 7.9, Marlowe's top rankings are dominated by the works of Shakespeare, with 17 of the 25 matches, and Jonson is hardly better, giving 14 of the 25 matches with Shakespeare; whereas in both cases the actual works of the author are not prominent, 5 by Marlowe and 7 by Jonson.

Marlowe Wrote Shakespeare?

It has been argued that Marlowe faked his death and used "Shakespeare" as his pen name to continue writing afterwards. However our results provide little evidence to support a particular relationship between the works of Marlowe and Shakespeare. It is true that plays by Marlowe tend to retrieve plays by Shakespeare, as shown in Table 7.9. However the evidence becomes much weaker when we compare results in Table 7.7 and Table 7.9 in detail. Sh.139 appears five times in five retrievals in Table 7.9. Given the hypothesis that the true author for this book is Marlowe, it should occasionally retrieve books by Marlowe. However, as can be seen in Table 7.7, when Sh.139 is used as a query, no works of Marlowe are retrieved. Sh.166 and Sh.147 share the same properties, that is, none of these query books retrieve works by Marlowe. The fact that Jonson's works also match those of Shakespeare also further suggests that the similarity with Marlowe might be a matter of period rather than authorship itself.

The results of positive leave-one-out classification experiments are also indicative, from the Table 7.3. The plays of Marlowe and Jonson we collect in the *Gutenberg634* data are never falsely attributed. To some extent this may be due to experimental design, where the presence of Shakespeare's plays in the negative examples is overwhelmed by the large volume of nineteenth-century text. However, on the other hand, in the negative leave-one-out experiments, the works of both Marlowe and Jonson are usually misattributed to Shakespeare; while those works of Beaumont & Fletcher are occasionally attributed to Shakespeare. In this respect, the error rate of false matches is high, indicating that the works of these authors cannot be distinguished well. Also there is no particular evidence to support the argument that the works of Shakespeare were actually written by any of these authors. Taking these

considerations and observations into account, we suggest that there is little evidence in our experiments to support the hypothesis that Marlowe wrote Shakespeare.

7.2.6 Beyond Precision: Authorship Search for Authorship Attribution

Besides the standard precision measure borrowed from IR, we reassessed the quality of attribution from another perspective on these results, using the search-based classification method introduced in Chapter 6. Rather than returning a list of rankings, an explicit attribution decision is made for each query book as an input, with which the authorship is to be identified. Here, the following rules are used. Given a query text by an unknown author, if three or more of the top 5 matches are by some author A', then we attribute the query text to A' with high confidence that the attribution is a correct assignment. Alternatively if two of the top 5 are by the author A' and the remainder are by three different authors, then we attributed the query text to A' with slightly lower confidence. Otherwise we have insufficient confidence to judge the authorship of the query book; the book remains unattributed.

Taking results of using function words for instance, we attribute with strong confidence correctly in 451 cases and incorrectly in 61 cases, giving an accuracy of 88.1%. We attribute with low confidence correctly in 20 cases and incorrectly in 23 cases. Attribution is unknown in 79 cases. Overall accuracy, equivalent to a recall value, is 74.3%; precision is 84.9%, measured by the number of query books being correctly attributed compared to the number of query books being attributed.

As can be seen, both classification-based models and search-based models are highly effective, given appropriate style markers. Using function words, the classification-based model achieves 85.6% overall accuracy in contrast to 74.3% of the search-based model. However, with search 84.9% precision is obtained with reasonably high confidence. The reported effectiveness is achieved by examining results in the top n rankings with $n = 5$, however higher confidence in attribution results can be obtained when $n < 5$, with a slight loss in recall potentially.

Importantly, unlike the leave-one-out experiments, which are effectively binary classification tasks, the search-based attribution is technically a multi-class classification, given 55 classes to be distinguished, which is dramatically harder than a binary classification. In

this respect the search-based attribution method is substantially more scalable than the classification-based attribution approach. The results are not perfect but dramatically better that random. Also, as demonstrated in Chapter 6, we have no reason to doubt that the effectiveness of the search-based model, both the precision and recall, can be boosted by a finer post-analysis of the retrieved top rankings, even using the same ranking mechanism. Improvement can be obtained by minimising either the number of unattributed query documents, or the number of misattributed query documents.

7.3 Summary

We have explored the effectiveness of authorship attribution on works of literature, based on the two types of approaches that have been proposed earlier in this thesis. The collection was newly derived from the *Project Gutenberg*, and named as *Gutenberg634*. Using such data, our results have shown that the positive leave-one-out classification can be highly effective, with an overall accuracy of over 85.6%, and the negative leave-one-out experiments have led to even more accurate results. On the other hand, the search-based attribution is in fact a multi-class AA of 55 classes, which is greatly harder than the leave-one-out investigation that is only binary classification. In this respect, even though the effectiveness of the search-based model was slightly lower than the classification-based model numerically, giving respective 74.3% recall and 84.9% precision, the scalability of AA was substantially increased in terms of the number of authors and the number of works being involved.

While these results are not comprehensive, they confirm that a majority of authors do indeed have an identifiable writing style. Moreover, the results also confirm that simple markers suffice to identify a particular author. Our best results are based on function words as markers of style; part-of-speech tags are reasonably effective, but are somewhat undermined by the fact that tagging is an error-prone process. The automated tagging tends to fail on text with unusual constructions, and such constructions are more likely to be indicative of distinguishing writing habits, whereas extraction of function words is straightforward and a lossless process.

Use of fragments of documents was less successful in contrast to the use of complete books. Experiments using 1,000-word fragments resulted in a clear failure to successfully

attribute works, while fragments of 10,000 words, somewhat over a tenth of a typical book, have been able to achieve correct attribution in over 50% of cases. This result is consistent with our previous exploration of AA on newswire data, where each article is typically much shorter than a book. Overall the accuracy of classification is much better than random, but is insufficient on its own to definitively determine authorship.

The series of experiments presented in this chapter allow us to understand some causes of attribution failure. The pattern of errors suggests that a key cause is a lack of distinct style in some texts, such as translated books. That is, some of the failures are due to properties of the works rather than weakness of the method itself. The exploration also allowed us to revisit the question of *Shakespeare authorship*; we did not discover strong evidence to support that these works were written by Marlowe.

A limitation of our experiments was that the sources were somewhat mixed; most of the works were from the nineteenth century; but a fraction was much older. Nonetheless, results were highly successful, and provided strong confirmation of the ability of simple statistical methods to effectively identify authorship.

Chapter 8

Conclusions and Future Work

Authorship attribution (AA) is the task of identifying authors of disputed or anonymous texts. It is concerned with style of writing rather than topic or subject matter. Broadly speaking, writing style can be viewed as the underlying methods of sentence construction that may be analysed by examination of a variety of elements, such as the richness of vocabulary, the sequences of words, the length of sentences, and the frequency of word usage. The accepted assumption behind AA is that each author writes in a distinct way; some writing characteristics cannot be manipulated by the writer's will, and thus can be identified by an automated process.

Techniques of AA are valuable for a wide range of applications, such as plagiarism detection in an academic environment, forensic investigations, and identification of the source of a piece of intelligence. The AA research area has a history of more than a hundred years; however, the achievements of computational AA are still unsatisfactory. The remaining challenges—that is, increased diversity in data sets, tokenising methods, classification methodologies, and evaluation design—have not resulted in consensus about the effectiveness of techniques. It is commonly the case that reported successes in AA research are subject to the specific terms set for a particular AA scenario, and thus may not be applicable to other AA tasks.

Based on a detailed review on the existing techniques and difficulties in AA—as described in Chapter 2—we have undertaken a series of AA investigations in this thesis, and presented our solutions to the reviewed issues. In this chapter, we summarise our primary research

contributions, discuss remaining issues, and consider directions for future work.

8.1 Research Contributions

Any authorship attribution approach starts with use of a set of training documents. The purpose of training is to learn an author's profile in terms of his or her writing habits. For a particular author, an accurate extraction of such a profile requires a sufficient volume of writing materials by that author. The profile is usually represented by style markers—that is, in-document features that are evidenced in the writing. The extracted style markers are then measured and differentiated by a classification process, where the classifier is learned on the profiles of the provided potential authors. Therefore, the quality of both the style markers and the classification methodologies has a strong impact on the effectiveness of AA.

In this research, we have developed a set of fully automated systems to improve the effectiveness, reliability, and scalability of AA techniques from different perspectives. In summary, our research makes the following contributions:

- We have developed a total of nine testbeds, drawn from two domains: newswire and English literature. Each has properties that makes it suitable for one or more types of AA investigations.

- We have investigated existing AA methods, using two of the newly-developed collections. This work has established the value of using our experimental framework, and tested whether the developed collections were able to distinguish different AA techniques.

- We have proposed a new KLD-based model inspired by information theory, which outperforms existing techniques for AA.

- We have examined the discrimination power of a variety of style marker types under a consistent experimental environment, including marker types that have been explored in earlier research and a range of markers that were proposed in this thesis.

- We further proposed three novel systems—the voting system, two-stage model prediction system, and the additive system—to combine evidence from different types of

markers. All three systems have been shown to be effective; each has its own advantages.

- To address one of the most challenging issues—that is, the scalability of previous AA approaches—we have proposed the novel task of authorship search (AS). This is, to the best of our knowledge, the first system that can effectively search for documents on an author-by-author basis.

- Authorship search was further proposed as a basis for AA, intended for large document collections. We evaluated this search-based AA method on multiple data sets, and our results have shown it to be effective and substantially scalable.

- In order to have a more comprehensive understanding of AA, in particular failed cases, we have compared both the classification-based and search-based KLD models on a collection of literature in English. The results have confirmed that our proposed methods are effective, robust, and scalable.

In the following sections, we detail the main discoveries that have been made in our research.

Establishing Testbeds and Baseline

Since the data sets used in earlier research are usually small, the reported success may not be reliable. Such doubt has motivated us to develop larger corpora for our investigations, since we believe that collections are an essential factor for AA evaluations. We developed a total of nine data sets for different types of AA investigations, including 2-class AA, n-class AA, one-class AA, authorship search, and search-based AA. The collections vary in size, to ensure that the scalability of techniques can be tested. Additionally, these collections are derived from different domains—newswire and English literature—so that the robustness of the proposed methods can be examined.

The initial AA investigation presented in Chapter 3 demonstrated the limitation of many earlier studies of AA. When the training data was small, 25 documents for instance, even a numerical difference in performance of over 10% is unlikely to be statistically significant. In addition, the effectiveness of any AA approach depends on the authors to be differentiated.

For example, the C4.5 decision tree algorithm produced 92.0% accuracy for binary AA between authors Currier and Beamish, but only 56.5% accuracy—that is, only slightly better than random—for Schweid and Beamish, when 25 training documents were involved. Similar trends were observed when the numbers of training samples were increased. The results indicated that it is not reliable to judge AA techniques by one or two specific tasks; the evaluation should be averaged across multiple runs. However, many prior AA investigations were concerned with specific author combinations, as reviewed in Chapter 2, and therefore it is not clear whether the reported success is applicable to other AA scenarios. In addition, the results suggested that the style markers used were effective for some cases, but not for others; this observation was reinforced in Chapter 5.

Our experiments with a popular collection, *The Federalist Papers*, demonstrated the limitations of many prior AA studies that used small data sets. With different methods, changes in only one or two attributions resulted in great differences numerically in terms of the overall accuracy; however, the differences were not statistically significant. The results also supported our argument that an evaluation based on small collections is unlikely to be reliable.

In this respect, all investigations presented in this thesis were based on collections with reasonable sizes, from over 5,000 to half a million documents. All experiments were undertaken in a consistent experimental framework to assure comparability, and with multiple runs to assure the reliability of the averaged effectiveness.

KLD-based Authorship Attribution Approach

In Chapter 4, a novel approach to AA was proposed. This approach was inspired by information theory, using Kullback-Leibler divergence—a measure of relative entropy—for classification. Language models incorporating smoothing techniques formed the basis of the model. We have shown in a series of experiments on multiple data sets that our KLD-based method is better than the best previous techniques, and it also has the advantages of simplicity and efficiency.

KLD-based AA method consistently outperformed Bayesian networks in all AA tasks, regardless of changes in the number of training samples. The improvements were shown to

be statistically significant. Compared to SVMs, our KLD-based model was superior when a relatively small number of training documents was used; otherwise, the SVMs performed slightly better, but the difference was marginal, and not statistically significant. Our KLD-based method can be directly applied to n-class AA, with any number of n; however, SVMs are limited when making classification to multiple classes, in that then need to convert an n-class problem into an 1-against-n classification—that is, binary AA. In this respect, our model is superior to SVMs.

In addition, our model is advantageous in terms of computational efficiency, with asymptotic cost that is almost linear in the number of training documents. In comparison, the best optimisation algorithm for SVMs has asymptotic cost of $O\left(kn^2\right)$, while the computation of a Bayesian network is even more expensive, increasing exponentially with the number of training documents and the number of cliques in the constructed networks.

Given the consistent classification methodology and selection of style markers, we also examined different probability approximation methods. The KLD-based model accommodates flexible ways of constructing the underlying language models. The computation of divergence is independent of the language models that are the methods for estimating probability distributions of style markers. We compared four smoothing techniques, to examine which method could lead to the best AA. However, we observed that with a sufficient number of training documents, there was no significant difference between choices of smoothing methods; when using a smaller training data set, the differences between optimally tuned systems was statistically significant, however. This is in agreement with our previous observations. The results suggest that the language models can be highly effective for estimating the probability distributions of style markers for AA.

Although the overall effectiveness of AA did not differ substantially with different smoothing techniques, the overall effectiveness was subject to the values of parameters in each smoothing method used. Some techniques—such as Dirichlet smoothing and two-stage smoothing—were greatly sensitive to parameter settings.

To test whether the query-centric smoothing (such as that used in IR) is sufficient for AA, we applied smoothing in two alternative ways—that is, estimating the style markers seen in the test documents, or all the pre-defined style markers including those that are absent in the

test documents. We found that considering both seen markers and unseen markers gave rise to more stable effectiveness, rather than using in-document markers only. This observation also indicated that the preference of not using certain markers can potentially contribute to the style definition.

Style Markers and Improvements

The style of writing is not easy to define or identify; researchers have proposed a variety of features to define writing style. To compare the goodness of these marker types for AA, we tested a series of style markers—including shallow linguistic features and deep linguistic features—under a consistent experimental design. This was to identify which was the best type of style marker for AA. We observed that there is no single set of style markers that was consistently superior for all AA tasks. Looking at the results on a case-by-case basis, some marker types were effective for some attribution tasks, but not for others.

From our investigation, we found that the style markers formed from deep linguistic parsing were time consuming and computationally costly. We had hoped that such features would lead to a better effectiveness, but surprisingly we did not observe promising results by using such information. The deep linguistic features did not guarantee a better effectiveness for AA, and overly complicated markers can lead to failure. These features are generally less effective than lexical features, particularly when only limited training data is provided. The extraction of such features is not lossless; it is usually the case that the most unusual usages of words or tokens cannot be parsed correctly due to the lack of instances for training, thus degrading AA performance.

As was demonstrated in Chapter 5, the effectiveness of AA based on only one single type of style marker was affected by various factors, such as the volume of training data, and the class of AA investigation. Given small training samples on binary AA for example, simpler style markers such as the function words were generally better. With larger numbers of training samples, and harder tasks (n-class AA), richer style markers can achieve better performance, such as the newly proposed FW/POS (function words with their lexical categories). Therefore, we suggest that the choice of style markers is task-dependent; a static set of pre-defined features can hardly satisfy all AA scenarios. Based on these observations, we

further explored the combination of style markers to improve AA.

By considering the trade-off between the computational efficiency and effectiveness, we have directed our research to the exploration of multiple types of features that were extracted from shallow linguistic parsing. We started this investigation with a very straightforward way of combining evidence—that is, increasing the size of the feature set. However this simple approach caused the effectiveness to decrease sharply. Some distinguishing features had a tendency to be overwhelmed by other features, and thus, effectiveness was severely degraded. To address this issue and effectively combine multiple types of style markers, we proposed three novel systems.

Model voting system: this approach is valuable when there are many existing techniques available. The integration of the existing approaches requires little modification to the base techniques; the output from the existing methods can be used as an input to the voting system. Voting can be based on existing techniques, but also on different types of style markers. The voting system is advantageous for its simplicity and effectiveness.

Two-stage model prediction system: this approach is less expensive in terms of computational cost. The results in Chapter 5 have shown 54% prediction accuracy, compared to 25% as expected based on random choice, given four types of features. In addition, in cases where the best prediction was not made, the second best feature type was usually predicted—we have observed that the differences between the best feature types were usually small. Further, our system has been shown to be able to avoid using particularly bad features for attribution. The two-stage prediction system is able to flexibly choose task-specific style markers for AA rather than relying on pre-defined style markers.

Additive modelling system: this is the most effective system of all the three systems, but more expensive in computational cost. Unlike the voting system, which is merely concerned with the number of votes that each author gets, the additive modelling system is also concerned with the size of each vote made for a particular author. Our additive modelling system has resulted in great improvement in the effectiveness of AA, in particular for harder n-class AA. For instance, as shown in Chapter 5, this system produced 90% accuracy for 5-

class AA, compared to 82.2% without using the additive modelling, a statistically significant difference.

Authorship Search and Search for Attribution

Authorship search (AS) and search-based AA were proposed for large document collections, to address one of the most critical issues in AA research: scalability. Our investigation was undertaken in multiple dimensions, and the results showed that AS is dramatically scalable in terms of the number of documents and number of potential authors involved (tens of thousands of documents, and several hundred authors).

In AS, we found that the proposed similarity measure—the Kullback-Leibler divergence—is far more effective than standard measures drawn from information retrieval. The KLD-based search model was able to effectively return documents that share the same author as the query documents; the highest precision was 44% with a collection of half a million documents, with contributions from more than 2,000 valid authors, as well as others. However, both the vector space model and the probabilistic Okapi model have shown severe failure, even with much smaller collections.

Whether topic-bearing words should be used for style analysis has been controversial. To provide some guideline and evidence, we have run AS using two indexing strategies: bag-of-words, and function words. The results have shown that the topic-bearing words are misleading in AS; little consistency in authorship was observed in the top-ranked documents. This result also indicated that in a large data collection—especially newswire data, where documents are written over a wide range of topics, but many are on the similar topics—it is better to use style markers that are free of content. As expected, the search effectiveness dropped when the size of the collection was increased. Indexing the larger collections with multiple types of style markers—both function words and part-of-speech tags—can significantly boost the effectiveness of AS, although part-of-speech tags were found to be much less effective on their own.

Methods proposed in prior AA research struggled to make accurate attributions to authors when given more than a few hundred documents or more than a few authors. We have shown that our AS system can be effectively used for AA. This search-based attribution method has

been effectively scaled to 10,000 documents, by several hundred authors. For example, the number of authors included in the APvote10k collection is nearly 342; more than 10% of the authors have over 100 contributions in the collection. This setup for authorship attribution is substantially more challenging than data used in any previous studies.

The search-based AA approach has been able to effectively distinguish the 342 authors. We evaluated 700 queries, each formed from a single article. The highest accuracy achieved was around 51%, and a further 10% improvement can be made by refining the analysis method; the expected correctness of random attribution is less than 0.3%. In addition, an overall effectiveness of 74% was obtained when the volume of query texts was increased—that is, concatenating 10 documents for each query. Our novel search-based attribution method has therefore been shown to be superior to existing techniques in both effectiveness and scalability.

Interestingly, we again observed strong inconsistencies between authors; queries for some authors, such as Currier and Dishneau, were much better than others, such as Skidmore. Our method was able to correctly distinguish Dishneau from the other 341 authors at 76% accuracy, but only 36% for Skidmore. With larger, 10-document queries, we have achieved 100% accurate attribution for Dishneau, but only 10% for Skidmore. Moreover, the 100 queries of Currier and Dishneau were attributed at 70% and 75% accuracy respectively, when we increased the number of documents in the collection, and the number of authors, by a factor of 10. Our next research was to understand why attribution sometimes fails.

When Attribution Fails

So far, most of our investigations were conducted using newswire data. However, articles in such collections are mostly short, and we do not expect human readers to be aware of strong styles of writing associated with certain authors. Therefore, in order to understand why attribution of authorship sometimes fails, we created another collection of English literature, consisting of 634 works from a total of 55 novelists who are renowned worldwide. Both classification-based and search-based models were used to attribute these works in multiple ways.

The pattern of errors shown in Chapter 7 suggested that a key cause of failure is the lack

of distinct style in some texts. For example, we have observed that one of the problematic authors was Tolstoy, for whom the attribution was always mismatched. We noted that most of the works were translated materials, indicating that the style did not survive in the translating process; sometimes there could be more than one translator involved in the process. In this respect, we argue that some of the failures were caused due to properties of the works or the lack of style for certain authors, rather than weakness of the proposed method itself.

The exploration also allowed us to revisit the question of the *Shakespeare authorship*; we did not discover strong evidence to support the argument that Marlowe has actually written Shakespeare plays.

8.2 Future Work

From our experience of research in the field of authorship attribution, we raise the following issues for the future work.

Consensus on Benchmarks. As we showed in Chapter 3, collections are essential for proper AA evaluations, and therefore, it is of great importance that researchers achieve a consensus on benchmarks in this area, as well as keeping a consistent experimental setup. The range of available AA techniques, together with the many potential applications, indicate that consensus on different benchmarks is needed.

Style Definition. In Chapter 5 we used several types of features to define the style of writing. Simple lexical features do not capture syntactic and semantic relationships from documents. One the other hand, richer features, such as syntax trees, are often expensive to extract, can be overly complicated, and do not occur frequently. In this case investigations based on such features were not particularly helpful. Therefore, how to effectively construct style markers from deep linguistic features remains an open challenge for future work.

In addition, the extraction of richer features is realised by natural language processing (NLP), meaning that the goodness of style markers and the effectiveness of AA are subject to the effectiveness of the NLP techniques used. Current NLP techniques require a

learning process; the most distinct writing patterns of certain authors may not be learned successfully due to the lack of training data. Therefore, minimising the information loss during NLP is also worth exploring.

Alternatively, as we have shown in Chapter 4, the rareness of word usage could also provide evidence of the writing style, and so may be helpful for identifying authorship. Therefore, improving AA by rare features could be interesting to investigate.

Combination of Features Although we have shown three highly effective AA systems to combine simple features and complicated features, the investigation was only preliminary, and further aspects should be examined.

In some cases, the voting system may not be plausible; the disadvantage is that such a system favors many votes for limited author candidates. For example, the voting system is less ideal in the circumstance where only two existing techniques are integrated to differentiate between three or more authors. Thus, how to choose the desirable values of l and δ for the potential authors remains a question, where l is the number of votes available, and δ is the threshold for making attributions. In our experiments, we set l and δ empirically; it will be stronger to establish a theoretical approach to identify values for l and δ given a number of authors.

For the two-stage model prediction system, more advanced learning approaches can be applied to improve the effectiveness of prediction. On the other hand, the prediction process in our experiments has a relatively high requirement regarding the volume of training data, which makes it less ideal for small collections. In this respect, a learning algorithm that is designed to learn from small samples would make the system more flexible. In addition, a trust level can be assigned to each prediction made; such a value can be used to adjust the computation of divergence.

For the additive system, we have equally weighted different types of style markers. However, the significance of marker types were clearly different, as observed. Therefore it is worth investigating methodologies that can effectively approximate the optimal values of each α_i, rather than using an arbitrary assignment.

Improving Search-based Authorship Attribution The style-search based AA methods have shown to be highly scalable and reasonably effective. Further improvement can be made in two directions: better ranking algorithms, and better post-ranking analysis. In our experiments, we observed that nearly 30% of instances cannot be attributed, and around 20% of instances were incorrectly attributed.

Unfortunately, our search system did not successfully scale to attribute the collection of 100,000 documents of thousands of authors. Improvements can be made by proposing better ranking algorithms; a top-ranked list with higher precision will certainly result in a more accurate assignment. Alteratively, a finer method for the post-ranking analysis would also be helpful. For example, given the same ranking algorithm and indexing method, a further 10% improvement was obtained when we took the ranks of each returned document into account.

To sum up, our research has made substantial contributions to the area of AA: our proposed new systems and models have substantially outperformed existing techniques, and an experimental framework has been established that allows for the consistent comparison of the effectiveness and scalability of future AA approaches.

Appendix A

The List of Selected Function Words

a	about	above	accordingly	across
after	afterwards	again	against	albeit
all	allow	allowable	allowed	allows
allowing	almost	alone	along	already
also	although	always	am	among
amongst	an	and	another	any
anybody	anyhow	anyone	anything	anywhere
apart	appear	appears	appropriate	are
aren	around	as	at	away
be	became	because	become	becomes
been	before	beforehand	behind	below
beside	besides	between	beyond	big
both	but	by	considering	cannot
co	consequently	consider	considerable	considered
can	considers	contain	containing	contains
corresponding	could	currently	did	didn
do	does	doesn	doing	done

down	downwards	dozen	during	each
eg	either	else	enough	entire
entirely	et	etc	even	ever
every	ex	example	except	exclusive
exclusively	far	few	first	firstly
for	former	forth	found	from
further	furthermore	get	given	go
gone	got	had	hadni	half
hardly	has	have	having	hence
here	hereafter	hereby	herein	hereupon
hitherto	how	howbeit	however	hundred
ie	if	immediate	in	inasmuch
inc	include	included	includes	including
indeed	indicate	indicated	indicates	inner
insofar	instead	into	inward	is
isn	it	its	itself	just
last	latter	latterly	least	less
lest	like	little	many	may
mean	meaning	means	meanwhile	might
missed	misses	missing	more	moreover
most	mostly	much	must	name
namely	near	necessary	neither	never
nevertheless	new	next	no	nobody
none	noone	nor	normally	not
note	notes	nothing	now	nowhere
of	off	often	oh	old
on	once	one	only	onto
or	other	others	otherwise	ought
out	outside	over	overall	own
particular	particularly	per	perhaps	placed

please	plus	possible	probably	provides
questionable	quite	rather	really	relatively
respectively	right	said	same	secondly
see	seem	seemed	seeming	seems
self	sensible	sent	serious	several
shall	should	shouldn	since	so
some	somebody	somehow	someone	something
sometime	sometimes	somewhat	somewhere	specified
specify	specifying	still	sub	such
sup	taken	than	that	the
then	thence	there	thereafter	thereby
therefore	therein	thereupon	these	this
thorough	thoroughly	those	though	thousand
through	throughout	thus	to	together
too	toward	towards	under	unless
until	unto	up	upon	use
used	useful	uses	using	usually
value	various	very	via	viz
vs	want	was	wasn	way
well	went	were	weren	what
whatever	when	whence	whenever	where
whereafter	whereas	whereby	wherein	whereupon
wherever	whether	which	while	whither
who	whoever	whole	whom	whose
why	will	with	within	without
would	wouldn	yet		

Appendix B

The List of Selected Brown Tags

Regular Tags	Description
ABL	pre-qualifier (quite, rather)
ABN	pre-quantifier (half, all)
ABX	pre-quantifier (both)
AP	post-determiner (many, several, next)
AT	article (a, the, no)
BE	be
BED	were
BEDZ	was
BEG	being
BEM	am
BEN	been
BER	are
BEZ	is
CC	coordinating conjunction (and, or)
CD	cardinal numberal (one, two, 2, etc.)
CS	subordinating conjunction (if, although)
DO	do
DOD	did

Regular Tags	Description
DOZ	does
DT	singular determiner/quantifier (this, that)
DTI	singular or plural determiner/quantifier (some, any)
DTS	plural determiner (these, those)
DTX	determiner/double conjunction (either)
EX	existential there
HV	have
HVD	had (past tense)
HVG	having
HVN	had (past participle)
IN	preposition
JJ	adjective
JJR	comparative adjective
JJS	semantically superlative adjective (chief,top)
JJT	morphologically superlative adjective (biggest)
MD	modal auxiliary (can, should, will)
NN	singular or mass noun
NN$	possessive singular noun
NNS	plural noun
NNS$	possessive plural noun
NP	proper noun or part of name phrase
NP$	possessive proper noun
NPS$	possessive plural proper noun
NR	adverbial noun (home, today, west)
OD	ordinal numeral (first, 2nd)
PN	nominal pronoun (everybody, nothing)
PN$	possessive nominal pronoun
PP$	possessive personal pronoun (my, our)
PP$$	second (nominal) possessive prounon (mine, ours)

Regular Tags	Description
PPL	singular reflexive/intensive personal pronoun (myself)
PPLS	plural reflexive/intensive personal pronoun (ourselves)
PPO	objective personal pronoun (me, him, it, them)
PPS	3rd. singular nominative pronoun (he, she, it, one)
PPSS	other nominative personal pronoun (I, we, they, you)
QL	qualifier (very, fairly)
QLP	post-qualifer (enough, indeed)
RB	adverb
RBR	comparative adverb
RBT	superlative adverb
RN	nominal adverb (here, then, indoors)
RP	adverb/particle (about, off, up)
TO	infinitive marker to
UH	interjection, exclamation
VB	verb, base form
VBD	verb, past tense
VBG	verb, present participle/gerund
VBN	verb, past participle
VBZ	verb, 3rd. singular present
WDT	wh-determiner (what, which)
WP$	possessive wh-pronoun (whose)
WPO	objective wh-pronoun (whom, which, that)
WPS	nominative wh-pronoun (who, which, that)
WQL	wh-qualifier (how)
WRB	wh-adverb (how, where, when)
ZZ	unknown

Hyphenated Tags	Description
FW-	foreign word, hypenated before regular tags
-NC	cited word, hyphenated after one or more regular tags
-HL	hyphenated to one or more regular tags of words in headlines
-TL	hyphenated to one or more regular tags of words in titles

Merge Tags	Description
+	used to join two or more regular tags
*	directly affixed to regular tags (not, n't)

Punctuation Tags	Description
.	sentence terminator (. ? !)
(	left paren
)	right paren
–	dash
,	comma
:	colon

Bibliography

D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.

G. Amati and C. J. V. Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems*, 20(4):357–389, 2002.

S. Argamon, M. Šarić, and S. S. Stein. Style mining of electronic messages for multiple authorship discrimination: First results. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 475–480, Washington, D.C., USA, 2003. ACM Press.

H. Baayen, H. V. Halteren, and F. Tweedie. Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution. *Literary and Linguist Computing Advance Access*, 11(3):121–132, 1996.

H. Baayen, H. V. Halteren, A. Neijt, and F. Tweedie. An experiment in authorship attribution. In *Proceedings 6th International Conference on the Statistical Analysis of Textual Data*, pages 29–37, 2002.

R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman, 1999.

J. Bai, D. Song, P. Bruza, J. Y. Nie, and G. Cao. Query expansion using term relationships in language models for information retrieval. In *Proceedings of the 14th ACM CIKM In-*

ternational Conference on Information Knowledge Management, pages 688–695, Bremen, Germany, 2005. ACM Press.

R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter. Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research*, 3:1183–1208, 2003.

D. Benedetto, E. Caglioti, and V. Loreto. Language trees and zipping. *The American Physical Society*, 88(4):048702, 2002.

Y. Bernstein and J. Zobel. A scalable system for identifying co-derivative documents. In *Proceedings of the 11th SPIRE String Processing and Information Retrieval Symposium*, pages 55–67, Padova, Italy, 2004. Springer.

J. N. G. Binongo. Who wrote the 15th book of Oz? An application of multivariate statistics to authorship attribution. *Computational Linguistics*, 16(2):9–17, 2003.

S. Bird. NLTK: The natural language toolkit. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL Interactive Presentation Sessions*, pages 69–72, Sydney, Australia, 2006. Association for Computational Linguistics.

A. Bookstein. Explanation and generalization of vector models in information retrieval. In *Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 118–132, West Berlin, Germany, 1982. Springer-Verlag New York.

G. Brajnik, S. Mizzaro, and C. Tasso. Evaluating user interfaces to information retrieval systems: a case study on user support. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 128–136, Zurich, Switzerland, 1996. ACM Press.

E. Brill. A simple rule-based part of speech tagger. In *Proceedings of the 3rd applied natural language processing*, pages 152–155, Trento, Italy, 1992. Association for Computational Linguistics.

W. Buntine. Theory refinement on bayesian networks. In *Proceedings of the 7th Annual Conference on Uncertainty Artificial Intelligence*, pages 52–60, Melbourne, Australia, 1991. Morgan Kaufmann Publishers.

W. Buntine. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, 8:195–210, 1996.

J. Burrows. Delta: A measure of stylistic difference and a guide to likely authorship. *Literary and Linguistic Computing*, 17:267–287, 2002.

J. Burrows. All the way through: Testing for authorship in different frequency strata. *Literary and Linguist Computing Advance Access*, 22(1):27–47, 2006.

J. Burrows. Word patterns and story shapes: the statistical analysis of narrative style. *Literary and Linguistic Computing*, 2:61–70, 1987.

J. Burrows. Computers and the study of literature. *Computers and Written Texts*, pages 167–204, 1992.

D. Canter and J. Chester. Investigation into the claim of weighted cusum in authorship attribution studies. *Forensic Linguistics*, 4(2):252–261, 1997.

C. Carin. The bard's fingerprints. *Lingua Franca*, 4:29–39, 1998.

C. E. Carole. Who's at the keyboard? authorship attribution in digital evidence investigations. *International Journal of Digital Evidence*, 4(1), 2005.

M. F. Caropreso, S. Matwin, and F. Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. *Text Databases and Document Management: Theory and Practice*, pages 78–102, 2001.

S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan. Using taxmonmy, discriminants and signatures for navigating in text database. In *Proceedings of the 23rd VLDB International Conference on Very Large Data Bases*, pages 446–455, Athens, Greece, 1997. Morgan Kaufmann Publishers.

C. C. Chang and C. J. Lin. *LIBSVM: A Library for Support Vector Machines*, 2001. http://www.csie.ntu.edu.tw/ cjlin/libsvm.

S. F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong. VideoQ: An automated content based video search system using visual cues. In *Proceedings of the 5th ACM MULTIMEDIA International Conference on Multimedia*, pages 313–324, Seattle, Washington, USA, 1997. ACM Press.

S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 310–318, Santa Cruz, California, USA, 1996. Morgan Kaufmanns Publishers.

T. S. Chua and L. Q. Ruan. A video retrieval and sequencing system. *ACM Transactions on Information Systems*, 13(4):373–407, 1995.

G. F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

W. S. Cooper. On selecting a measure of retrieval effectiveness. *Journal of the American Society for Information Science*, 24:87–100, 1973.

T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

R. M. Coyotl-Morales, L. Villasenor-Pineda, M. M. y Gómez, and P. Rosso. Authorship attribution using word sequences. In *Proceedings of the 11th CIARP Iberoamerican Congress on Pattern Recognition*, pages 844–853, Cancun, Mexico, 2006. Springer.

W. B. Croft and J. Lafferty. *Language Modeling for Information Retrieval*. Kluwer Academic Publishers, Norwell, MA, USA, 2003.

P. De-Haan. Review of Farringdon, "Analysing for Authorship: A Guide to Cusum Technique". *Forensic Linguistics*, 5(1):69–76, 1998.

J. Diederich, J. Kindermann, E. Leopold, and G. Paass. Authorship attribution with support vector machines. *Applied Intelligence*, 19(1-2):109–123, 2003.

P. Domingos and M. J. Pazzani. On the optimality of the simple bayesian classifier under zerone loss. *Machine Learning*, 29(2/3):103–130, 1997.

H. Drucker, V. Vapnik, and D. Wu. Automatic text categorization and its applications to text retrieval. *IEEE Transactions on Neural Networks*, 10(5):1048–1054, 1999.

D. D'Souza, J. Thom, and J. Zobel. Collection selection for managed distributed document databases. *Information Processing and Management*, 40:527–546, 2004.

S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the 7th ACM CIKM International Conference on Information Knowledge Management*, pages 148–155, Bethesda, Maryland, USA, 1998. ACM Press.

B. Efron and R. Thisted. Estimating the number of unseen species: How many words did shakespeare know? *Biometrika*, 63(3):435–447, 1976.

J. M. Farringdon. *Analysing for Authorship: A Guide to the Cusum Technique*. University of Wales Press, 1996.

D. Foster. *Author Unknown: On the Trail of Anonymous*. Henry Holt, New York, 2000.

W. B. Frakes. Stemming algorithms. *Information Retrieval: Data Structures and Algorithms*, pages 131–160, 1992.

W. B. Frakes and C. J. Fox. Strength and similarity of affix removal stemming algorithms. *ACM SIGIR Forum*, 37(1):26–30, 2003.

N. Friedman and M. Goldszmidt. Building classifiers using bayesian networks. In *Proceedings of the 13th National Conference on Artificial Intelligence*, pages 1277–1284, Portland, Oregon, 1996a. American Association for Artificial Intelligence Press.

N. Friedman and M. Goldszmidt. Discretization of continuous attributes while learning bayesian networks. In *Proceedings of the 13th ICML International Conference on Machine Learning*, pages 157–165, Bari, Italy, 1996b. Morgan Kaufmann Publishers.

N. Friedman and M. Goldszmidt. Learning Bayesian networks with local structure. In *Proceedings of the NATO Advanced Study Institute on Learning in graphical models*, pages 421–459, Erice, Italy, 1998. Kluwer Academic Publishers.

D. Frietag and A. K. McCallum. Information extraction with HMMs and shrinkage. In *Proceedings of AAAI-99 Workshop on Machine Learning for Information Extraction*, pages 31–36, Menlo Park, California, USA, 1999. American Association for Artificial Intelligence Press.

N. Fuhr and C. Buckley. A probabilistic approach for document indexing. *ACM Transactions on Information Systems*, 9(3):223–248, 1991.

G. Fung. The disputed Federalist papers: SVM feature selection via concave minimization. In *Proceedings of 2003 Conference on Diversity in Computing*, pages 42–46, Atlanta, Georgia, USA, 2003. ACM Press.

E. Gabrilovich and S. Markovitch. Text categorization with many redundant features: Using aggressive feature selection to make SVMs competitive with C4.5. In *Proceedings of the 21st ICML International Conference on Machine Learning*, pages 321–328, Banff, Alberta, Canada, 2004. ACM Press.

J. F. Gao, J. Y. Nie, G. Y. Wu, and G. H. Cao. Dependence language model for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 170–177, Sheffield, UK, 2004. ACM Press.

G. Gaughan, A. F. Smeaton, C. Gurrin, H. Lee, and K. McDonald. Design, implementation and testing of an interactive video retrieval system. In *Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 23–30, Berkeley, California, USA, 2003. ACM Press.

J. Goodman. Extended comment on language trees and zipping, 2002. URL http://citeseer.ist.psu.edu/goodman02extended.html.

N. Gövert, M. Lalmas, and N. Fuhr. A probabilistic description-oriented approach for categorising Web documents. In *Proceedings of the 8th ACM CIKM International Conference on Information Knowledge Management*, pages 475–482, Kansas City, Missouri, USA, 1999. ACM Press.

R. A. Hardcastle. Cusum: a credible method for the determination of authorship? *Science and Justice*, 37(2):129–138, 1997.

D. Harman. Overview of the second text retrieval conference (TREC-2). *Information Processing and Management*, 31(3):271–289, 1995.

D. Heckerman. A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Redmond, Washington, 1995, Revised 1996. URL http://research.microsoft.com/research/pubs/view.aspx?msr_tr_id=MSR-TR-%95-06.

D. Heckerman, D. Geiger, and D. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

W. Hersh, C. Buckley, T.Leone, and D. Hickman. Ohsumed: An interative retrieval evaluation and new large text collection for research. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 192–201, Dublin, Ireland, 1994. ACM Press.

D. Hiemstra. Term-specific smoothing for the language modeling approach to information retrieval: the importance of a query term. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 35–41, Tampere, Finland, 2002. ACM Press.

D. I. Holmes. The analysis of literary style: a review. *Royal Statistical Society (Series A)*, 148(4):328–341, 1985.

D. I. Holmes. Authorship attribution. *Computers and the Humanities*, 28(2):87–106, 1994.

D. I. Holmes. The evolution of stylometry in humanities scholarship. *Literary and Linguistic Computing*, 13(3):111–117, 1998.

D. I. Holmes and R. S. Forsyth. The Federalist revisited: New directions in authorship attribution. *Literary and Linguistic Computing*, 10(2):111–127, 1995.

D. I. Holmes, M. Robertson, and R. Paez. Stephen Crane and the New York Tribune: A case study in traditional and non-traditional authorship attribution. *Computers and the Humanities*, 35(3):315–331, 2001.

J. Hoorn, S. Frank, W. Kowalczyk, and F. D. Ham. Neural network identification of poets using letter squences. *Literary and Linguistic Computing*, 14(3):311–338, 1999.

D. L. Hoover. Statistical stylistics and authorship attribution: An empirical investigation. *Literary and Linguistic Computing*, 16:421–444, 2001.

F. Jelinek and R. Mercer. Interpolated estimation of markov source parameters from sparse data. In *Proceedings of an International Workshop on Pattern Recognition in Practice*, pages 381–402, Amsterdam, The Netherlands, 1980. North-Holland Publisher.

T. Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *Proceedings of the 14th ICML International Conference on Machine Learning*, pages 143–151, Nashville, Tennessee, USA, 1997. Morgan Kaufmann Publishers.

T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th ECML European Conference on Machine Learning*, pages 137–142, Chemnitz, Germany, 1998. Springer.

G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence*, pages 338–345, Montreal, Quebec, Canada, 1995. Morgan Kaufmann Publisher.

K. S. Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. *Information Processing and Management*, 36 (6):779–840, 2000.

P. Juola. What can we do with small corpora? Document categorization via cross-entropy. In *Proceedings of the Interdisciplinary Workshop on Similarity and Categorization*, Edinburgh, UK, 1997.

P. Juola and H. Baayen. A controlled-corpus experiment in authorship identification by cross-entropy. *Literary and Linguistic Computing*, 20:59–67, 2003.

P. Juola, J. Sofko, and P. Brennan. A prototype for authorship attribution studies. *Literary and Linguist Computing Advance Access*, 21:169–178, 2006.

D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, 2000.

A. Kaster, S. Siersdorfer, and G. Weikum. Combining text and linguistic document representations for authorship attribution. In *SIGIR Workshop on Stylistic Analysis of Text for Information Access*, pages 27–35, Salvador, Bahia, Brazil, 2005. ACM Press.

F. Keulen. The Dutch computer corpus pilot project. In *Corpus Linguistics II: New Studies in the Analysis and Exploitation of Computer Corpora*, pages 127–162, Amsterdam, The Netherlands, 1986. Rodopi.

V. Kešelj, F. C. Peng, N. Cercone, and C. Thomas. N-gram-based author profiles for authorship attribution. In *Proceedings of PACLING Pacific Association for Computational Linguistics*, pages 256–264, Halifax, Nova Scotia, Canada, 2003.

D. V. Khmelev and W. J. Teahan. Comment on "language trees and zipping". *Physical Review Letters*, 90(8), 2003a. URL http://link.aps.org/abstract/PRL/v88/e048702.

D. V. Khmelev and W. J. Teahan. A repetition based measure for verification of text collections and for text categorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 104–110, Toronto, Canada, 2003b. ACM Press.

D. V. Khmelev and F. Tweedie. Using markov chains for identification of writers. *Literary and Linguistic Computing*, 16(4):229–307, 2002.

B. Kjell. Authorship attribution of text samples using neural networks and bayesian classifiers. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 1660–1664, San Antonio, Texas, 1994a. IEEE Press.

B. Kjell. Authorship determination using letter pair frequencies with neural network classifiers. *Literary and Linguistic Computing*, 9(2):119–124, 1994b.

B. Kjell and O. Frieder. Visualization of literary style. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 656–661, Chicago, Illinois, USA, 1992. IEEE Press.

D. Klein and C. D. Manning. Fast exact inference with a factored model for natural language parsing. In *Proceedings of the 15th NIPS Advances in Neural Information Processing Systems*, pages 3–10, Vancouver, British Columbia, Canada, 2002. MIT Press.

R. Klinkenberg and T. Joachims. Detecting concept drift with support vector machines. In *Proceedings of the 17th ICML International Conference on Machine Learning*, pages 487–494, Stanford, California, USA, 2000. Morgan Kaufmann Publishers.

A. Kolcz, V. Prabakarmurthi, and J. Kalita. Summarization as feature selection for text categorization. In *Proceedings of the 10th ACM CIKM International Conference on Information Knowledge Management*, pages 365–370, Atlanta, Georgia, USA, 2001. ACM Press.

M. Koppel and J. Schler. Exploiting stylistic idiosyncrasies for authorship attribution. In *Exploiting Stylistic Idiosyncrasies for Authorship Attribution. In IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis*, Acapulco, Mexico, 2003.

M. Koppel and J. Schler. Authorship verification as a one-class classification problem. In *Proceedings of the 21st ICML International Conference on Machine Learning*, pages 489–495, Banff, Alberta, Canada, 2004. ACM Press.

O. V. Kukushkina, A. A. Polikarpov, and D. V. Khmelev. Using literal and grammatical statistics for authorship attribution. *Problem of Information Transmission*, 37(2):172–184, 2001.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

J. T. Kwok. Automated text categorization using support vector machine. In *Proceedings of the 5th ICONIP International Conference on Neural Information Processing*, pages 347–351, Kitakyushu, Japan, 1998. IOA Press.

J. Lafferty and C. X. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 111–119, New Orleans, Louisiana, USA, 2001. ACM Press.

Y. S. Lai and C. H. Wu. Meaningful term extraction and discriminative term selection in text categorization via unknown-word methodology. *ACM Transactions on Asian Language Information Processing*, 1(1):34–64, 2002.

P. Langley and S. Sage. Tractable average-case analysis of naïve Bayesian classifiers. In *Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence*, pages 220–228, Montreal, Quebec, Canada, 1999. Morgan Kaufmann Publisher.

L. S. Larkey and W. B. Croft. Combining classifiers in text categorization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 37–50, Zurich, Switzerland, 1996. ACM Press.

M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2(1):1–19, 2006.

D. D. Lewis. Representation and learning in information retrieval, Amherst, US. *PhD Thesis*, 1992a.

D. D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 37–50, Copenhagen, Denmark, 1992b. ACM Press.

D. D. Lewis. Naïve bayes at forty: The independence assumption in information retrieval. In *Proceedings of the 10th ECML European Conference on Machine Learning*, pages 4–15, Chemnitz, Germany, 1998. Springer-Verlag.

D. D. Lewis and M. Ringuette. Comparison of two learning algorithms for text categorization. In *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, Las Vegas, Nevada, USA, 1994.

D. D. Lewis, Y. M. Yang, T. G. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

H. Li and K. Yamanishi. Text classification using ESC-based stochastic decision list. In *Proceedings of the 8th ACM CIKM International Conference on Information Knowledge Management*, pages 122–130, Kansas City, Missouri, USA, 1999. ACM Press.

J. Y. Li, M. S. Sun, and X. Zhang. A comparison and semi-quantitative analysis of words and character-bigrams as features in chinese text categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 545–552, Sydney, Australia, 2006. Association for Computational Linguistics.

T. Li, S. H. Zhu, and M. Ogihara. Efficient multi-way text categorization via generalized discriminant analysis. In *Proceedings of the 12th ACM CIKM International Conference on Information Knowledge Management*, pages 317–324, New Orleans, Louisiana, USA, 2003. ACM Press.

X. Y. Liu and W. B. Croft. Passage retrieval based on language models. In *Proceedings of the 11th ACM CIKM International Conference on Information Knowledge Management*, pages 375–382, McLean, Virginia, USA, 2002. ACM Press.

X. Y. Liu and W. B. Croft. Cluster-based retrieval using language models. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 186–193, New York, NY, USA, 2004. ACM Press.

A. Lohrey. Linguistics and the law. *Polemic*, 2(2):74–76, 1991.

K. Luyckx and W. Daelemans. Shallow text analysis and machine learning for authorship attribution. In *Proceedings of the CLIN 15th Meeting of Computational Linguistics in the Netherlands*, pages 149–160, University of Leiden, The Netherlands, 2005. The Netherlands Graduate School of Linguistics.

K. Luyckx, W. Daelemans, and E. Vanhoutte. Stylogenetics: Clustering-based stylistic analysis of literary corpora. In *Proceedings of LREC 2006 Workshop on Towards Computational Models of Literary Analysis*, Genova, Italy, 2006.

D. Mackay and L. Peto. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1(3):289–307, 1995.

M. B. Malyutov. Authorship attribution of texts: a review. In *Proceedings of the Program "Information Transfer"*, pages 1–17, University of Bielefeld, Germany, 2004.

C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

T. Masuyama and H. Nakagawa. Two step POS selection for SVM based text categorization. *IEICE Transactions on Information System*, E87-D(2):373–379, 2004.

A. McCallum and K. Nigam. A comparison of event models for naïve bayes text classification. In *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48, 1998a.

A. K. McCallum and K. Nigam. Employing EM in pool-based active learning for text classification. In *Proceedings of the 15th ICML International Conference on Machine Learning*, pages 350–358, Madison, Wisconsin USA, 1998b. Morgan Kaufmann Publishers.

M. Melucci. Context modeling and discovery using vector space bases. In *Proceedings of the 14th ACM CIKM International Conference on Information Knowledge Management*, pages 808–815, Bremen, Germany, 2005. ACM Press.

M. Melucci. Ranking in context using vector spaces. In *Proceedings of the 15th ACM CIKM International Conference on Information Knowledge Management*, pages 866–867, Arlington, Virginia, USA, 2006. ACM Press.

T. C. Mendenhall. The characteristic curves of composition. *Science*, 9:237–249, 1887.

J. Mitchell. *Who Wrote Shakespeare?* Thames & Hudson, 1996.

S. Mizzaro. Relevance: The whole history. *Journal of the American Society for Information Science*, 49(9):810–832, 1997.

S. Mizzaro. How many relevance in information retrieval? *Interacting with Computers*, 10 (3):303–320, 1998.

D. Mladenic and M. Grobelnik. Feature selection for classification based on text hierarchy. Working notes of learning from text and the Web, conference on automated learning and discovery, 1998.

R. D. Mori and F. Brugnara. HMM methods in speech recognition. *Survey of the State of the Art in Human Language Technology*, pages 21–30, 1996.

A. Moschitti and R. Basili. Complex linguistic features for text classification: A comprehensive study. In *Proceedings of the 26th ECIR European Conference on Information Retrieval*, pages 181–196, Sunderland, UK, 2004. Springer-Verlag.

F. Mosteller and D. Wallace. *Inference and Disputed Authorship: The Federalist.* Addison-Wesley Publishing Company, 1964.

H. Ney. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech and Language*, 8:1–38, 1994.

N. Oostdijk. *Corpus Linguistics and the Automatic Analysis of English.* Rodopi, 1991.

F. C. Peng, D. Schuurmans, and S. Wang. Language and task independent text categorization with simple language models. In *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics*, pages 110–117, Edmonton, Canada, 2003a. Association for Computational Linguistics.

F. C. Peng, D. Schuurmans, S. J. Wang, and V. Kešelj. Language independent authorship attribution using character level language models. In *Proceedings of the 10th conference*

on European chapter of the Association for Computational Linguistics, pages 267–274, Budapest, Hungary, 2003b. Association for Computational Linguistics.

M. S. Pol. A stylometry-based method to measure intra and inter-authorial faithfulness for forensic applications. In *SIGIR Workshop on Stylistic Analysis of Text for Information Access*, Salvador, Bahia, Brazil, 2005. ACM Press.

J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, Melbourne, Australia, 1998. ACM Press.

M. F. Porter. An algorithm for suffix stripping. *Readings in Information Retrieval*, 14(3): 130–137, 1980.

R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

A. C. Rencher. *Methods of multivariate analysis*. New York : J. Wiley, 2002.

S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

S. E. Robertson, C. J. Van-Rijsbergen, and M. F. Porter. Probabilistic models of indexing and searching. In *Proceedings of the 3rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 35–56, Cambridge, England, 1980. Butterworth.

S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau. Okapi at TREC. In *Proceedings of Text Retrieval Conference (TREC)*, pages 21–30, Gaithersburg, Maryland, USA, 1992.

J. Rudman. The state of authorship attribution studies: some problems and solutions. *Computers and the Humanities*, 31:351–365, 1998.

G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):323–328, 1988.

G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1984.

C. Sanderson and S. Guenter. On authorship attribution via markov chains and sequence kernels. In *Proceedings of the 18th ICPR International Conference on Pattern Recognition*, pages 437–440, Hong Kong, 2006. IEEE Computer Society.

A. Sarkar, A. D. Roeck, and P. H. Garthwaite. Term re-occurrence measures for analyzing style. In *SIGIR Workshop on Stylistic Analysis of Text for Information Access*, Salvador, Bahia, Brazil, 2005.

M. Sassano. Virtual examples for text classification with support vector machines. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 208–215, Sapporo, Japan, 2003. Association for Computational Linguistics.

L. Schamber. Relevance and information behavior. *Annual Review of Information Science and Technology*, 29:3–48, 1994.

B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.

H. Schütze, D. Hull, and J. O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 229–237, Seattle, Washington, USA, 1995. ACM Press.

F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

F. Sebastiani, A. Sperduti, and N. Valdambrini. An improved boosting algorithm and its application to automated text categorization. In *Proceedings of the 9th ACM CIKM International Conference on Information Knowledge Management*, pages 78–85, McLean, Virginia, United States, 2000.

W. Q. Shang, H. K. Huang, H. B. Zhu, Y. M. Lin, Y. L. Qu, and Z. H. Wang. A novel feature selection algorithm for text categorization. *Expert Systems with Applications*, 33 (1):1–5, 2007.

C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.

X. H. Shen, B. Tan, and C. X. Zhai. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–50, Salvador, Brazil, 2005. ACM Press.

A. Singhal and G. Salton. Automatic text browsing using vector space model. In *Proceedings of Dual-Use Technologies and Applications Conference*, pages 318–324, Utica, New York, USA, 1995.

A. Singhal, G. Salton, M. Mitra, and C. Buckley. Pivoted document length normalization. *Information Processing and Management*, 32(5):619–633, 1996.

M. W. A. Smith. Recent experience and new developments of methods for the determination of authorship. *Association for Literary and Linguistic Computing Bulletin*, 11:73–82, 1983.

I. Soboroff and C. Nicholas. Collaborative filtering and the generalized vector space model. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 351–353, Athens, Greece, 2000. ACM Press.

E. Sormunen. Liberal relevance criteria of TREC: counting on negligible documents? In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 324–330, Tampere, Finland, 2002. ACM Press.

E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Automatic authorship attribution. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, pages 158–164, Bergen, Norway, 1999. Association for Computational Linguistics.

E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Automatic text categorization in terms of genre and author. *Computational Lingustics*, 26(4):471–495, 2000.

E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Computer-based authorship attribution without lexical measures. *Computers and the Humanities*, 35(2):193–214, 2001.

K. Storey. Forensic text analysis. *Law Institute Journal*, 67(2):1176–1178, 1993.

Y. Suga, N. Kosugi, and M. Morimoto. Real-time background music monitoring based on content-based retrieval. In *Proceedings of the 12th ACM MULTIMEDIA International Conference on Multimedia*, pages 120–127, New York City, New York, USA, 2004. ACM Press.

D. L. Swets and J. Y. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):831–836, 1996.

S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2002.

Y. H. Tseng. Content-based retrieval for music collections. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 176–182, Berkeley, California, USA, 1999. ACM Press.

F. J. Tweedie, S. Singh, and D. I. Holmes. Neural network applications in stylometry: The Federalist papers. *Computers and the Humanities*, 30:1–10, 1996.

K. Tzeras and S. Hartmann. Automatic indexing based on bayesian inference networks. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 22–35, Pittsburgh, Pennsylvania, USA, 1993. ACM Press.

O. Uzuner and B. Katz. Style versus expression in literary narrative. In *SIGIR Workshop on Stylistic Analysis of Text for Information Access*, Salvador, Bahia, Brazil, 2005. ACM Press.

V. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, 1995.

V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

O. D. Vel, A. Anderson, M. Corney, and G. Mohay. Mining e-mail content for author identification forensics. *Special Interest Group on Management of Data Record*, 30(4): 55–64, 2001.

B. Wang and S. Zhang. A novel text classification algorithm based on naïve bayes and KL-divergence. In *Proceedings of the 6th International Conference on Parallel and Distributed Computing Applications and Technologies*, pages 913–915, Dalian, China, 2005. IEEE Computer Society.

R. Weber and M. Mlivoncic. Efficient region-based image retrieval. In *Proceedings of the 12th ACM CIKM International Conference on Information Knowledge Management*, pages 69–76, New Orleans, Louisiana, USA, 2003. ACM Press.

C. Williams. Word-length distribution in the works of Shakespeare and Bacon. *Biometrika*, 62:207–212, 1975.

I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools with Java Implementations*. Morgan Kaufmann Publishers, 2000.

I. H. Witten, A. M., and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers, 1999.

M. Wolters and M. Kirsten. Exploring the use of linguistic features in domain and genre classification. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, pages 142–149, Bergen, Norway, 1999. Association for Computational Linguistics.

S. K. M. Wong, W. Ziarko, V. V. Raghavan, and P. C. N. Wong. On modeling of information retrieval concepts in vector space. *ACM Transactions on Database System*, 12(2):299–321, 1987.

Y. M. Yang. A study on thresholding strategies for text categorization. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 137–145, New Orleans, Louisiana, USA, 2001. ACM Press.

Y. M. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1-2):69–90, 1999.

Y. M. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49, Berkeley, California, USA, 1999. ACM Press.

Y. M. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th ICML International Conference on Machine Learning*, pages 412–420, Nashville, Tennessee, USA, 1997. Morgan Kaufmann Publishers.

Y. M. Yang, J. Zhang, and B. Kisiel. A scalability analysis of classifiers in text categorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 96–103, Toronto, Canada, 2003. ACM Press.

G. U. Yule. On sentence-length as a statistical characteristic of style in prose, with applications to two cases of disputed authorship. *Biometrika*, 30:363–390, 1938.

C. X. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 334–342, New Orleans, Louisiana, USA, 2001a. ACM Press.

C. X. Zhai and J. Lafferty. Model-based feedback in the KL-divergence retrieval model. In *Proceedings of the 10th ACM CIKM International Conference on Information Knowledge Management*, pages 403–410, Atlanta, Georgia, USA, 2001b. ACM Press.

C. X. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, 2004.

D. Zhang and W. S. Lee. Extracting key-substring-group features for text classification. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 474–483, Philadelphia, PA, USA, 2006. ACM Press.

Y. Zhao and P. Vines. Authorship attribution via combination of evidence. In *Proceedings of the 29th ECIR European Conference on Information Retrieval*, pages 661–669, Rome, Italy, 2007. Springer.

Y. Zhao and J. Zobel. Effective authorship attribution using function word. In *Proceedings of the 2nd AIRS Asian Information Retrieval Symposium*, pages 174–190, Jeju Island, South Korea, 2005. Springer.

Y. Zhao and J. Zobel. Search with style: authorship attribution in classic literature. In *Proceedings of the 30th ACSC Australasian Computer Science Conference*, pages 59–68, Ballarat, Australia, 2007a. CRIPT.

Y. Zhao and J. Zobel. Authorship search in large document collections. In *Proceedings of the 29th ECIR European Conference on Information Retrieval*, pages 381–392, Rome, Italy, 2007b. Springer.

Y. Zhao, J. Zobel, and P. Vines. Using relative entropy for authorship attribution. In *Proceedings of the 3rd AIRS Asian Information Retrieval Symposium*, pages 92–105, Singapore, 2006. Springer.

J. Zobel and A. Moffat. Inverted files for text search engines. *ACM Computing Surveys*, 38: 1–56, 2006.

J. Zobel and A. Moffat. Exploring the similarity space. *ACM SIGIR Forum*, 32(1):18–34, 1998.